RM-76-76

ON SOME MULTI-SITE MULTI-SEASON STREAMFLOW GENERATION MODELS

Janusz Kindler

Wlodzimierz Zuberek

December 1976

Research Memoranda are interim reports on research being conducted by the International Institute for Applied Systems Analysis, and as such receive only limited scientific review. Views or opinions contained herein do not necessarily represent those of the Institute or of the National Member Organizations supporting the Institute.

PREFACE

In accordance with the suggestions of the IIASA Planning Conference held in June 1973 and with subsequent discussions with IIASA National Member Organizations, the IIASA Water Project (presently the Water Group of the Resources and Environment Area) concentrated during the years 1974-1975 on specific problems of an universal methodology for planning, design and operation of water resource systems.

Taking into account the importance of streamflow generation models for the design and operation of complex water resource systems, a special study was undertaken on "Intercomparison and improvement of existing stochastic models of multi-site and multi-season streamflow generation".

This paper describes the results of in-house research concerned with the comparison of three models and with the development of a computer package for multi-site multi-season streamflow generation.

SUMMARY*

The relative performance of some multi-site multi-season models is compared with respect to their adequacy for simulating monthly streamflow sequences. The three models brought under examination are the extended version of the multi-variate model proposed by Matalas (1967), the model formulated by Young and Pisano (1968), and the disaggregation model of Valencia and Schaake (1972). Computer implementation of these models has been accomplished in the form of the Multi-site Multi-season Streamflow Generation Package (MMSGP). Evaluation and comparison of the models has been carried out in terms of statistical flow parameters only. Some of these parameters are not explicitly built into the model structure. At the end, some general comments concerning applicability of each model are presented.

---

*Since this phase of investigations was completed, Mejia and Roussele (1976) have proposed modification of the disaggregation model which will be taken into account in the further work on the MMSGP.

TABLE OF CONTENTS

		Page
	Preface	iii
	Summary	v
1.	Introduction	1
2.	The Autoregressive Sequential, and the Disaggregation Streamflow Generation Models	3
3.	The investigations and their results	11
3.1	Normalizations of historic streamflow sequences	12
3.2	Estimation of the sample statistics	14
3.3	Solution of the $BB^T = C$ or $EE^T = F$ equations	15
3.4	Comparison of the sequential, and the disaggregation models	15
4.	Conclusions	18
	Acknowledgments	20
	List of Figures	21
APPENDIX:	COMPUTER IMPLEMENTATION	33
1.	Introduction	33
2.	File Structures	34
3.	General description of Input Parameters	35
4.	Main Segment	36
5.	Main Subroutines	36
5.1	Subroutine SEL	36
5.2	Subroutine TRF	37
5.3	Subroutine MAM	37
5.4	Subroutine MYM	38
5.5	Subroutine MAG	38
5.6	Subroutine DIM	38
5.7	Subroutine DIS	39
5.8	Subroutine RWC	39
5.9	Subroutine AGG	39
5.10	Subroutine EST	40
5.11	Subroutine OUT	40
5.12	Subroutine FDA	40
5.13	Subroutine FFA	41

TABLE OF CONTENTS continued.

6. Examples 41
6.1 Example 1: Generation 41
6.2 Example 2: Disaggregation 43

COMPUTER PROGRAM 45

1. Introduction

Over the last decade it has been generally recognized that digital simulation is usually almost the only technique which can be practically applied for design or analysis of a complex water resources system. The major reason simulation is so attractive for such studies is the great generality of the problem formulation to which it can be applied. Nonlinearities in the system equations can easily be handled. Constraints on state variables introduce no difficulty, stochastic effects can be taken into account.

Typically, the data available on the stochastic nature of hydrologic system inputs consists of a limited set of observations. Very rarely is there considered to be a sufficient period of record available to span all possible ways that the river flow might occur. It is known, however, that there is a large amount of information contained in recorded flow data that is not effectively used when simulation is based solely on the historical streamflow sequences. To overcome this inadequacy, the concept of streamflow synthesis has been introduced about 15 years ago by the pioneering works of Thomas and Fiering (1962) in the USA and of Svanidse (1964) in the USSR. Today, stochastic techniques of streamflow synthesis or generation are referred to as synthetic or operational hydrology. These techniques enable the planner to subject alternative water resources system designs to a set of synthetic streamflow consequences, each of which statistically resembles the historical one. Simulation of the system operation based on a large number of equally likely synthetic streamflow sequences, provides the planner with a means of estimating the expected risks and losses associated with a particular design of a water resources system.

Autoregressive Markovian generation models have an important place in the theory of stochastic modelling of streamflow sequences, and they also are most commonly used in practical applications. This refers specially to modelling of seasonally varying processes occurring at several locations in the river basin. Water management is essentially spatial in concept and generation models which accommodate both temporal (serial) and spatial (cross) correlation of river flows are needed in most of the actual planning projects.

This report is concerned with the multivariate streamflow generation model originally formulated by Matalas (1967) and the disaggregation model proposed by Valencia and Schaake (1972). The objective of this research was to compare the performance of these models with respect to their adequacy for simulating monthly streamflow sequences. The stationary model of Matalas was adopted for generation of monthly flows in two versions. The first of them follows the proposal of Bernier (1971), while the second one is based on the approach advised by Young and Pisano (1968). Computer implementation of all models has been accomplished in form of a Multi-site Multi-season Streamflow Generation Package (MMSGP). The MMSGP can be used for sequential generation of annual and seasonal streamflow sequences (in principle any subdivision of a year) as well as for disaggregation of annual synthetic sequences into the seasonal ones. The investigations were limited to lag-one Markovian model of both mean annual and mean monthly streamflow events, and they do not cover analysis of long-run dependencies (e.g. Hurst phenomenon). In accordance with the results of many investigations (e.g. Yevjevich, 1964) the second order stationarity of annual streamflow series is assumed. Describing the models, all flow sequences are assumed to be standardized with zero mean and standard deviation of one. The capital letters denote the matrices.

Three sets of historical mean monthly streamflow records from Canada (26 years, 3 sites), Czechoslovakia (40 years, 4 sites) and Poland (25 years, 4 sites) were used for testing consecutive versions of the MMSGP. The final version presented in this report is operationally correct, although a considerable number of various problems encountered during investigations require further work. Most of these problems are rather typical for all studies on synthetic hydrology. It is hoped that their orderly discussion may throw some light on future research in this area.

2. The Autoregressive Sequential, and the Disaggregation Streamflow Generation Models

For the generation of annual synthetic flows at n sites, the multivariate stationary lag-one Markov model (Matalas, 1967) may be written as:

$$X_i = A \cdot X_{i-1} + B \cdot \varepsilon_i \tag{1}$$

$$\text{for } i = 1,\ldots,z$$

where X_i and X_{i-1} are (n x 1) matrices whose elements are the annual flows at all sites in years i and i-1 respectively. The z denotes the desired length of synthetic sequences (number of years). The ε_i is an (n x 1) matrix of random components that are Nrm (0,1) distributed and independent of X_{i-1}. The (n x n) A and B matrices specify the time and space interdependence of flows. They are estimated from the historic sequences in such a way that the multivariate synthetic sequences generated by application of equation (1) will resemble the historic sequences in terms of the mean values, standard deviations, and lag-one serial, lag-one cross and lag-zero cross correlation coefficients of the annual flows. The elements of matrices A and B are estimated through solving the following matrix equations:

$$A = R_1 \cdot R_o^{-1} \tag{2}$$

$$BB^T = R_o - R_1 \cdot R_o^{-1} \cdot R_1^T \tag{3}$$

where R_o and R_1 are the lag-zero and lag-one annual correlation matrices respectively.

Derivation of equations (2) and (3) may be found in the above quoted work of Matalas (1967) as well as in the well known monograph of Anderson (1958) on the multivariate statistical analysis.

The elements of the R_o and R_1 matrices can be estimated as follows:

$$R_o = \left[r_o(k,l) \right] = \frac{1}{p} \sum_{i=1}^{p} x(k,i) \cdot x(l,i) \qquad (4)$$

for k,l = 1,...,n

$$R_1 = \left[r_1(k,l) \right] = \frac{1}{p-1} \sum_{i=2}^{p} x(k,i) \cdot x(l,i-1) \qquad (5)$$

for k,l = 1,...,n

It can be seen from equation (5) that common assumption of circular series is not employed in the study. It has been ascertained that in case of small sample size comparatively minor change in the flow sequence distort the estimation of lag-one correlation coefficients significantly.

The statistical analysis of the monthly flow sequences involves a consideration of stationarity not generally a problem in annual flow series. The monthly sequences are composed of values from 12 different populations, which fact accounts for their non-stationarity. Since a theory for non-stationary processes is practically nonexisting, Young and Pisano (1967) applied to the multivariate case the single site residual method of Yevjevich (1966) to achieve stationarity in the mean and standard deviation of the monthly flow series. This was accomplished by subtracting the appropriate monthly mean from the actual flow and dividing the result by the appropriate monthly standard deviation (standardization). Following estimation of R_o and R_1 matrices from the residual series, equation (1) was used for generation of synthetic residuals which next were destandardized into synthetic monthly flows. It is known, however, that removal of non-stationarity in the mean and standard deviation is not sufficient to achieve the second-order stationarity. Such an approach presumes that the seasonal fluctuation in the lag-one correlation coefficients can be ignored (O'Connell, 1972).

Another approach (Bernier, 1971), takes explicitly into account usually highly significant variation of correlation between the flows of two successive months. Taking into account the cyclic character of monthly flow sequences, the model is then the set of 12 regression equations which may be written in general form as:

$$X_{i,t} = A_t \cdot X_{j,u} + B_t \cdot \varepsilon_{i,t} \tag{6}$$

where

$$12i + t = 12j + u + 1 \quad ;$$
$$t,u = 1,\ldots,12 \quad ;$$
$$i,j = 1,\ldots,z$$

where $X_{i,t}$ and $X_{j,u}$ are the (n x 1) matrices whose elements are the monthly flows at all sites in months t and u of years i and j respectively. The $\varepsilon_{i,t}$ is an (n x 1) matrix of random components that are Nrm (0,1) distributed and independent of $X_{j,u}$. The elements of each pair of the (n x n) matrices A_t and B_t are estimated through solving similar equations as (2) and (3) which take form:

$$A_t = R_{1,t} \cdot R_{o,t-1}^{-1} \tag{7}$$

$$B_t B_t^T = R_{o,t} - R_{1,t} \cdot R_{o,t-1}^{-1} \cdot R_{1,t}^T \tag{8}$$

where $R_{o,t}$ and $R_{o,t-1}$ are lag-zero monthly correlation matrices in months t and t-1 respectively. The $R_{1,t}$ is lag-one monthly correlation matrix, whose elements are lag-one serial correlation coefficients of monthly flows at each site and lag-one cross correlation coefficients of monthly flows at different sites. The synthetic sequences resemble the historic sequences in terms of mean values, standard deviations, lag-one serial, lag-zero cross, and lag-one cross correlation coefficients of monthly flows.

The elements of the $R_{o,t}$ (or $R_{o,t-1}$) and $R_{1,t}$ can be estimated as follows:

$$R_o = \left[R_{o,t}\right] = \left[r_o(k,l,t)\right] = \frac{1}{p}\sum_{i=1}^{p} x(k,i,t) \cdot x(l,i,t) \qquad (9)$$

for $k,l = 1,\ldots,n$;
$t = 1,\ldots,12$;

$$R_1 = \left[R_{1,t}\right] = \left[r_1(k,l,t)\right] = \frac{1}{p-1}\sum_{i=2}^{p} x(k,i,t) \cdot x(l,i,t-1) \qquad (10)$$

for $k,l = 1,\ldots,n$;
$t = 1,\ldots,12$;
$x(l,i,t-1) = x(l,i-1,12)$ for $t = 1$;

where $x(k,i,t)$, $x(l,i,t)$ and $x(l,i,t-1)$ are the monthly flows at site k in year i and month t, at site l in year i and month t, and at site l in year i and month (t-1) respectively.

In 1972, Valencia and Schaake have formulated the model for disaggregation of the synthetic sequences of annual flows into synthetic sequences of seasonal flows (quarterly, monthly, etc.). According to the authors, the disaggregation model has two major advantages. First one is that it may be applied in conjunction with any of the presently existing models for sequentially generating annual events. The non-Markovian models, like FGN (Mandelbrot and Wallis, 1969) and Broken Line (Mejia, Rodriguez-Iturbe and Dawdy, 1972) can be applied only to the generation of sequences associated with stationary processes such as annual flows. The synthetic annual sequences generated by these models (taking care of long-run flow dependencies) can be step by step disaggregated into streamflow sequences corresponding to smaller time intervals. Because of the computational difficulties with matrices of higher order, the annual flows are usually first disaggregated into quarterly values, and next quarterly flows are disaggregated into monthly flows. Of course other sequence of disaggregation may be followed as well. Major advantage of the model is that at each level of disaggregation, the flow sequences maintain the relevant statistics of the higher level (e.g. the average of mean monthly flows of a given quarter equals the mean quarterly flow, the average of mean quarterly flows equals the mean annual flow, etc.).

The equation for disaggregation of annual flows into quarterly flows (Valencia and Schaake , 1972) is

$$Y_i = D \cdot X_i + E \cdot \varepsilon_i \qquad (11)$$

$$\text{for } i = 1, \ldots, z$$

where X_i is an (n x 1) matrix whose elements are the already generated synthetic annual flows at all sites in year i. The Y_i is an (4n x 1) matrix whose elements are the quarterly flows at all sites in year i. The ε_i is an (4n x 1) matrix of random components that are Nrm (0,1) distributed. The (4n x n) D matrix and (4n x 4n) E matrix specify the time and space interdependence of annual and quarterly flows. The elements of these matrices are estimated through solving the following matrix equations:

$$D = R_{yx} \circ R_x^{-1} \qquad (12)$$

$$EE^T = R_y - R_{yx} \circ R_x^{-1} \circ R_{yx}^T \qquad (13)$$

where $R_x = R_o$ (see equation (4)). The (4n x 4n) R_y matrix specify all "within-the-year" temporal and space correlation dependencies of quarterly flows. The (4n x n) R_{yx} matrix specify correlation dependencies between annual flows and the corresponding quarterly flows at all sites. Similar to equations (2),(3),(7), and (8), equations (12) and (13) are again based on the theory of the multivariate normal distribution.

The elements of the R_y and R_{yx} matrices can be estimated as follows:

$$R_y = \left[r_y(n(t-1)+k, n(u-1)+l) \right] = \frac{1}{p} \sum_{i=1}^{p} y(k,i,t) \cdot y(l,i,u) \quad (14)$$

for $t,u = 1,\ldots,4$;
$k,l = 1,\ldots,n$;

$$R_{yx} = \left[r_{yx}(n(t-1)+k, l) \right] = \frac{1}{p} \sum_{i=1}^{p} y(k,i,t) \cdot x(l,i) \quad (15)$$

for $t = 1,\ldots,4$;
$k,l = 1,\ldots,n$;

where $y(k,i,t)$ and $y(l,i,u)$ are the quarterly flows at site k in quarter t of year i, and at site l in quarter u of year i respectively. The $x(l,i)$ is the standardized annual flow at site l in year i.

According to Valencia and Schaake (1972) it is said that thus generated quarterly sequences "will resemble, in terms of the expected values of the seasonal statistics, the historical samples". These statistics are the means and variances at the different stations (sites), the correlation between seasonal values at the same station or different stations, and the correlation between the seasonal value at any station and the annual value at any station." It can be seen, however, from equation (14) that the disaggregation model does not take into account the correlation dependencies between quarterly flows in the last quarter of year i and the first quarter of year i+1.

The equation for disaggregation of quarterly flows into monthly flows is:

$$Y_{i,s} = D_s \cdot X_{i,s} + E_s \cdot \varepsilon_{i,s} \quad (16)$$

for $i = 1,\ldots,z$;
$s = 1,\ldots,4$;

where $X_{i,s}$ is an (n x 1) matrix whose elements are the quarterly flows at all sites in quarter s of year i. The $Y_{i,s}$ is an (3n x 1) matrix whose elements are the monthly flows at all sites in quarter s of year i. The $\varepsilon_{i,s}$ is an (3n x 1) matrix of random components that are Nrm (0,1) distributed. The (3n x n) D_s matrix and (3n x 3n) E_s matrix specify the space and time interdependence of quarterly flows in quarter s and the corresponding monthly flows. The elements of these matrices are estimated through solving the following matrix equations:

$$D_s = R_{yx,s} \cdot R_{x,s}^{-1} \tag{17}$$

$$E_s E_s^T = R_{y,s} - R_{yx,s} \cdot R_{x,s}^{-1} \cdot R_{yx,s}^T \tag{18}$$

where $R_{x,s}$ is an (n x n) matrix that specifies all cross correlation dependencies of quarterly flows in quarter s. The (3n x 3n) $R_{y,s}$ matrix specifies all "within-the-quarter" temporal and space correlation dependencies of monthly flows in quarter s. The (3n x n) $R_{yx,s}$ matrix specifies correlation dependencies between quarterly flows and the corresponding monthly flows at all sites.

Each of the $R_{x,s}$ matrices is built of the appropriate elements of the quarterly correlation matrix estimated according to equation (14). The elements of $R_{x,s}$, $R_{y,s}$ and $R_{yx,s}$ martices can be estimated as follows:

$$R_x = \left[R_{x,s} \right] = \left[r_x(k,l,s) \right] =$$

$$= \frac{1}{p} \sum_{i=1}^{p} x(k,i,s) \cdot x(l,i,s) \tag{19}$$

$$\text{for} \quad s = 1,\ldots,4 \quad ; \quad k,l = 1,\ldots,n$$

$$R_y = \left[R_{y,s}\right] = \left[r_y(n(t-1)+k, n(u-1)+l, s)\right] =$$

$$= \frac{1}{p} \sum_{i=1}^{p} y(k,i,3(s-1)+t) \cdot y(l,i,3(s-1)+u) \qquad (20)$$

for $s = 1,\ldots,4$;
$t,u = 1,\ldots,3$;
$k,l = 1,\ldots,n$

$$R_{yx} = \left[R_{yx,s}\right] = \left[r_{yx}(n(t-1)+k, l, s)\right] =$$

$$= \frac{1}{p} \sum_{i=1}^{p} y(k,i,3(s-1)+t) \cdot x(l,i,s) \qquad (21)$$

for $s = 1,\ldots,4$;
$t = 1,\ldots,3$;
$k,l = 1,\ldots,n$

where $y(k,i,3(s-1)+t)$ and $y(l,i,3(s-1)+u)$ are the monthly flows at site k in month $3(s-1)+t$ of year i and at site l in month $3(s-1)+u$ of year i respectively. The $x(l,i,s)$ is the quarterly flow at site l in quarter s of year i.

Quoting again Valencia and Schaake (1972)--"the monthly traces thus generated will preserve the following monthly statistics: means and variances, correlation between any two monthly values within a season (quarter) and any seasonal value in this season." It can be seen also from equation (20) that the disaggregation model does not take into account the correlation dependencies between monthly flows in the last month of quarter s and the first month of quarter s+1.

The number of disaggregation operations at the quarterly level is equal to the number of years in the synthetic annual sequences. At the monthly level, the number of disaggregation operations is four times higher.

To summarize, the Multi-site Multi-season Streamflow Generation Package (computer implementation of the above presented models) have been used in the study reported herein for:

* sequential generation of annual flow sequences according to equation (1),

* sequential generation of monthly flow sequences according to equation (1), following "stationarization" of the process by removal of non-stationarity in the means and standard deviations,

* sequential generation of monthly flow sequences according to equation (6),

* disaggregation of annual flow sequences, generated by application of equation (1), into quarterly flows (eq.(11)) and next into monthly flows (eq.(16)).

Since one of the aims of the study was to check if the sequentially generated monthly flow sequences hold the historic statistics associated with the higher level of aggregation, the MMSGP is provided with the subroutine aggregating monthly flows into quarterly and annual values. At each level of aggregation mean values, standard deviations, skewness, kurtosis and all lag-zero and lag-one correlation coefficients are estimated Their comparison could be effected by the statistical tests, however, the authors have restrained themselves to the qualitative analysis only. One of the reasons is that most of the available tests are developed for the statistics drawn from the normally distributed samples. The historical samples as well as the synthetic samples do not satisfy this requirement. Another reason is that for historic sample sizes usually available in hydrology such tests have very low power.

3. The investigations and their results

Working on the MMSGP, the authors have encountered a number of different problems which are discussed in this section of the report. First of all they pertain to normalization of historic streamflow sequences, estimation of sample statistics, solution of the $BB^T = C$ or $EE^T = F$ equations and the choice of adequate criteria for comparison of sequential and disaggregation models of monthly streamflow sequences.

3.1 Normalization of historic streamflow sequences

The hydrological literature contains many references to the properties of various probability distribution functions in fitting streamflow records. Unfortunately, a typical history of flows is quite short, between 10 and 50 years, and consequently the statistical tests available for testing the goodness of fit of theoretical distributions to large quantities of empirical data must be applied with great care. The selection of a distribution must involve some intuition and common sense (Fiering and Jackson, 1971). A similar conclusion was reached by Trykozko (1973) who analysed the possibilities of determination of a non-parametric test for selection of a distribution type. On the basis of a very extensive experimental material, Trykozko underlines that in case of the small-size samples ($N < 50$), differentiation of alternative hypotheses concerning distribution type is always highly problematic.

As far as the distribution of mean monthly flows is concerned, the log-normal and Pearson-type III distributions are probably most popular. In case of the two-parameter log-normal distribution, transformation $y = \ln(x)$ changes the sequence $\{x\}$ of natural flows into the sequence $\{y\}$ of the normally distributed flows. If the historical flows are assumed to follow a three-parameter log-normal distribution, normalization of the process involves among others estimation of the lower bound of the variable. The method of moments and the method of maximum likelihood were both tried to obtain the estimates of all distribution parameters, but it was found that the lower bounds are negative in most of the analysed cases. Since negative lower bound is not compatible with the physical properties of streamflow processes, the three-parameter log-normal distribution was not used in the investigations reported in this paper. For the Pearson Type III distribution, transformation $y = \sqrt[3]{x}$ leads to the approximately normally distributed flows.*) Another transformation which is

*) Instead of $y = \sqrt[3]{x}$, Kaczmarek (1970) has shown that transformation $y = x^{0.28}$ gives better results in the case of that distribution.

sometimes applied without making reference to any particular distribution is $y = \sqrt[2]{x}$.

In light of the above mentioned difficulties associated with determination of the distribution of natural flows, it was decided to incorporate in the MMSGP four normalization options (no transformation, natural logarithm, square root, cube root) and to develop some criteria for the selection of transformation that brings natural flows closest to the normal distribution. All considerations in this section are based on the generally acknowledged hypothesis that normalization of the marginal distributions leads to the approximately normal multivariate variable.

However, development of an adequate and easy for computer implementation criteria of normality proved to be difficult. One of the contemplated tests was to be based on comparison of the values of skewness and kurtosis estimated for the sequences "normalized" by application of different transformations. This is illustrated by some of the results presented in Figures 1 and 2. Very similar results were obtained for other flow data. It has been noticed that all transformations reduce the skewness and kurtosis close to the required values of zero and three respectively, but it is very difficult to indicate which transformation is the best one.*)

Finally all the above listed transformation options were incorporated in the MMSGP, but selection of the appropriate one was left to the decision of the program user. All further investigations reported in this paper have been limited to the sequences "normalized" by the logarithmic transformation. The assumption that the natural mean monthly flows follow the log-normal distribution is generally believed to be acceptable, and it could not be proved that some other distribution will better fit the flow data used for the investigations reported in this paper. An important advantage of the logarithmic transformation is also that the synthetic flows in the generated sequences cannot be negative.

---

*) Some of these difficulties are due to the fact that different normalization options are most effective for different months of the same set of historical record.

3.2 Estimation of the sample statistics

Assuming the monthly flows to be log-normally distributed, the means, standard deviations and correlation coefficients which are used for determination of the A and B (or D and E) matrices can be estimated either directly from the logarithms of historical events or by application of the well known equations (Aitchison and Brown, 1957) relating the statistics of the normal and log-normal distributions. To preserve the historical statistics of flows rather than that of their logarithms, Matalas (1967) recommends application of the second of these two possibilities. Consequently, an attempt was made to employ in the MMSGP equations which relate parameters of the normal and log-normal distributions. It has been found, however, that in a good number of cases this approach leads to the difficulties in the solution of the $BB^T = C$ (or $EE^T = F$) equations. The matrix BB^T (or EE^T) should be positive definite what is a necessary requirement in order for B (or E) to be real. Unfortunately, this condition could not be always satisfied. The reason is, that the relations between population statistics of the log-normal and normal distribution, do not necessarily hold for the sample statistics. This might be especially true in case of the sample sizes usually available in hydrology.

Under these circumstances, it was decided to compute the A and B (or D and E) matrices on the basis of statistics estimated from the historical sequences following their "normalization". In addition to the reasons presented above, it should be noted that equations relating parameters of the normal and other than log-normal distributions are not readily available. However, the most important reason is that the procedure finally adopted for computation of the A and B (or D and E) matrices always leads to a positive definite estimate for BB^T (or EE^T), as proved by Valencia and Schaake (1972).

Another and probably even more fundamental question pertains to the reliability of statistics estimated from the usually short historic streamflow sequences. Here comes the question of standard errors and biases associated with the sample statistics, the population values of which are unknown. It has been ascertained many times in this study, that the estimated parameters are highly variable even for a minor change of the sample size. More detailed discussion of this problem falls outside the scope of the present paper, however, further advancement of stochastic hydrology requires a considerable research effort in this area. One of the possible approaches might be the Bayesian inference providing finite sample probability distribution function for the unknown parameters.

3.3 Solution of the $BB^T = C$ or $EE^T = F$ equations

As it is known there is no unique solution of equation $BB^T = C$ (or $EE^T = F$) and the usual procedure to determine matrix B (or E) is by application of the principal component technique or by introduction of an assumption that B (or E) is a lower triangular matrix. In the MMSGP first of these techniques is applied, since it seems to be the more general one. It should be noted, however, that comparative computations have not revealed superiority of any one of these techniques.

3.4 Comparison of the sequential, and the disaggregation models

Development of general criteria for evaluation and comparison of generation models seems to be one of the critical and still unresolved issues in synthetic hydrology. In fact one may wonder if development of universally acceptable criteria is a feasible task at all. Taking into consideration the operational sense of synthetic hydrology, this should be probably rather a set of rules to be followed depending on the type of the water resources problem to be solved by simulation over the synthetic streamflow sequences.

At present two approaches to this problem are most common. First of them is based on the comparative analysis of some statistical parameters of the historical and synthetic flows. One may look at the model performance in terms of parameters that the model was or was not explicitly constructed to preserve.

Comparison of the Markovian models reported herein has been carried out in terms of some statistical parameters of the historic and synthetic flow sequences. The authors realize that such analysis cannot be fully conclusive, but it was intended to give at least an account what certainly cannot be expected from particular models. The statistical parameters subject to comparison were averaged over the set of generated synthetic sequences.

All types of generating models discussed in section 2 of this report were programmed as a Multi-site Multi-season Streamflow Generation Package (MMSGP) and applied to the available monthly flow series. First the historical monthly flows were aggregated into annual flows and equation (1) was used to generate a set of 100 years long, synthetic sequences of annual events. The results of statistical examination of these sequences generated on the basis of the Polish and Czechoslovak flow data are shown in Figs. 3 and 4. It can be seen that there is a good agreement between the corresponding statistics of historical and synthetic sequences. Now one of the major questions to be answered by our investigations was if the sequentially generated monthly flows maintain the relevant annual statistics of the historical record. The synthetic monthly flows were generated by application of equation (6). Following aggregation into annual events they were brought under examination and the results are shown together with the other annual statistics in Figs. 3 and 4. Although there is a good reproduction of historical means, standard deviations and lag-zero cross correlation coefficients, the sequentially generated monthly flows do not maintain the lag-one serial and lag-one cross sorrelation coefficients at the annual level.

In the next step, the annual events already generated by application of equation (1) were disaggregated first into quarterly (equation 11) and next into monthly (equation 16) flows. It was intended to examine if the monthly flows derived by disaggregation preserve the relevant monthly statistics of the historical record. The results of the analysis are displayed in Figs. 5,6,7, and 8 just for two months of November and December (the first and the second month of the hydrological year), for two sets of Polish and Czechoslovak flow records. It can be seen that there is a satisfactory resemblance of all December statistics, however, the November flows derived by disaggregation do not maintain the historical lag-one correlation dependencies. Although it is not shown here, results of the computations have indicated that February, May and August flows exhibit the similar lack of resemblance with the historical record. Referring to section 2 of this report, it should be underlined again that the disaggregation model does not take into account the correlation dependencies between quarterly flows in the last quarter of one year and the first quarter of the following year. At the second level of disaggregation the correlation dependencies between monthly flows in the last month of each quarter and the first month of the following quarter are also not taken into account. This is fully confirmed by the results of our computations, part of which is shown graphically in Figs. 5,6,7, and 8. The results of statistical examination of the historical and sequentially generated (equation (6)) monthly sequences are shown in the same figures. Although there are some discrepancies due to the fact that the synthetic statistics were averaged over the relatively modest set of synthetic sequences, there is evident resemblance of statistics estimated on the basis of historical and sequentially generated sequences. An example of additional comparison of historical and sequentially generated (equation (6)) monthly sequences is shown in Fig. 9. The curves show the per cent of time that the mean monthly flow at a given site is smaller than given amounts regardless of continuity in time. Good correspondence of such curves developed both for historical and synthetic samples was ascertained for all sites and all historical data sets.

Since the preceding analysis was concerned only with flow frequency, regardless of flow continuity in time, another attempt was made to compare the number and the duration (length) of flow series whose elements are (1) less or equal or (2) higher or equal than some preselected levels. The reduced flow levels (see note in Fig. 9) were chosen to be 0.2, 0.5, 1.0, 1.5, 2.0, and 3.0; each of these numbers is followed by the sign "+" or "-" as to indicate flows $\geq$ or $\leq$ than the given level. The duration (length) of the flow series was analyzed for a maximum of 15 consecutive time periods (months). In Figs. 10 and 11 some results of this analysis are presented. The cumulative frequency curves of reduced flow series refer to four sites (one historical and five synthetic flow sequences for each) and to the flow levels of 0.5- (Fig. 10) and 1.0+ (Fig. 11). Similar agreement between such curves was ascertained for all other flow levels of the historical and synthetic flow sequences subject to analysis.

The results of monthly flow generation by application of equation (1) to the "stationarized" historical record (Young and Pisano, 1967) are not shown in this report. It was ascertained, however, that there is a lack of resemblance between the correlational structure of historical and synthetic flows, both at the monthly and annual levels. In the MMSGP computer coding of this model is free of errors as noted by O'Connell (1973) as well as by Finzi, Todini and Wallis (1974).

None of the models discussed in this report takes explicitly into account the coefficient of skewness or higher order moments of the distribution. However, the MMSGP output gives also the values of the coefficient of skewness and kurtosis, both for the historical and synthetic sequences. No comparison of these statistics was attempted since they exhibit considerable variability from one synthetic sequence to the other generated by the same model.

4. Conclusions

The main conclusions drawn from the investigations are as follows:

(1) Multi-site sequential generation of monthly flows which assumes that the process is lag-one Markovian, nonstationary and cyclic (equation (6)), yields reasonable results with good resemblance of historical record in terms of the means, standard deviations as well as the cross and serial correlational structure at the monthly level. The synthetic monthly sequences are consistent with the historical pattern of annual flows in terms of their means, standard deviations and lag-zero cross correlation coefficients. It should be underlined that these statistics are not explicitly built into the model structure. The lag-one correlational structure of the historical annual flows

is not preserved. It seems, however, that if the synthetic sequences are to be used for simulation of the water resources systems providing seasonal (within-the-year) storage only, this lack of resemblance should not be of much importance in evaluating alternate designs.

(2) Multi-site sequential generation of monthly flows which assumes that the process is lag-one Markovian and approximately stationary after removal of non-stationarity in the means and standard deviations, seems to be the least satisfactory technique. This model may be applied only if the correlational structure of historical flows do not exhibit month-to-month variability but such situations are quite unusual.

(3) Generation of synthetic monthly flows by disaggregation of the previously generated annual flows, as proposed by Valencia and Schaake (1972), raises some doubts. If a particular time step is adopted for simulation, the highest priority resemblance with historical record should apply to the statistics referring specifically to this time step. Unfortunately, at each level of disaggregation 25% of lag-one serial and lag-one cross correlation coefficients are not preserved.

(4) There are many difficulties associated with development of the multi-site multi-season streamflow generation models and considerable research effort is needed in this area. Most of these difficulties may be attributed to the shortness of hydrological records and instability of small samples. As noted by many, existing streamflow records are not sufficiently extensive to provide reliable estimates of many statistics "important" for a proper design of water resources systems. One of the possible ways out of this dilemma seems to be expansion of investigations concerning the mechanism underlying the physical generation of river flows. But at the same time more investigations attempting to assess which parameters really are "important" for a proper design of a water resources system--how sensitive is the design to changes in these parameters--seems to be necessary.

Acknowledgments

The authors wish to express their gratitude to Professor Zdzislaw Kaczmarek, IIASA Water Project Leader, for his continued advice and support during all phases of the study. Acknowledgment is also due to the participants of the IIASA Workshop on Multi-site Streamflow Generation Models (Laxenburg, February 1976) for their comments on the preliminary version of this report.

LIST OF FIGURES

1. Polish Rivers - C_s values of natural flow sequences and sequences "normalized" by logarithmic and cube root transformations.

2. Polish Rivers - Kurtosis of natural flow sequences and sequences "normalized" by logarithmic and cube root transformations.

3. Comparison of statistical parameters of annual flow sequences (Polish Rivers).

4. Comparison of statistical parameters of annual flow sequences (Czechoslovak Rivers).

5. Comparison of statistical parameters of November flows (Polish Rivers).

6. Comparison of statistical parameters of December flows (Polish Rivers).

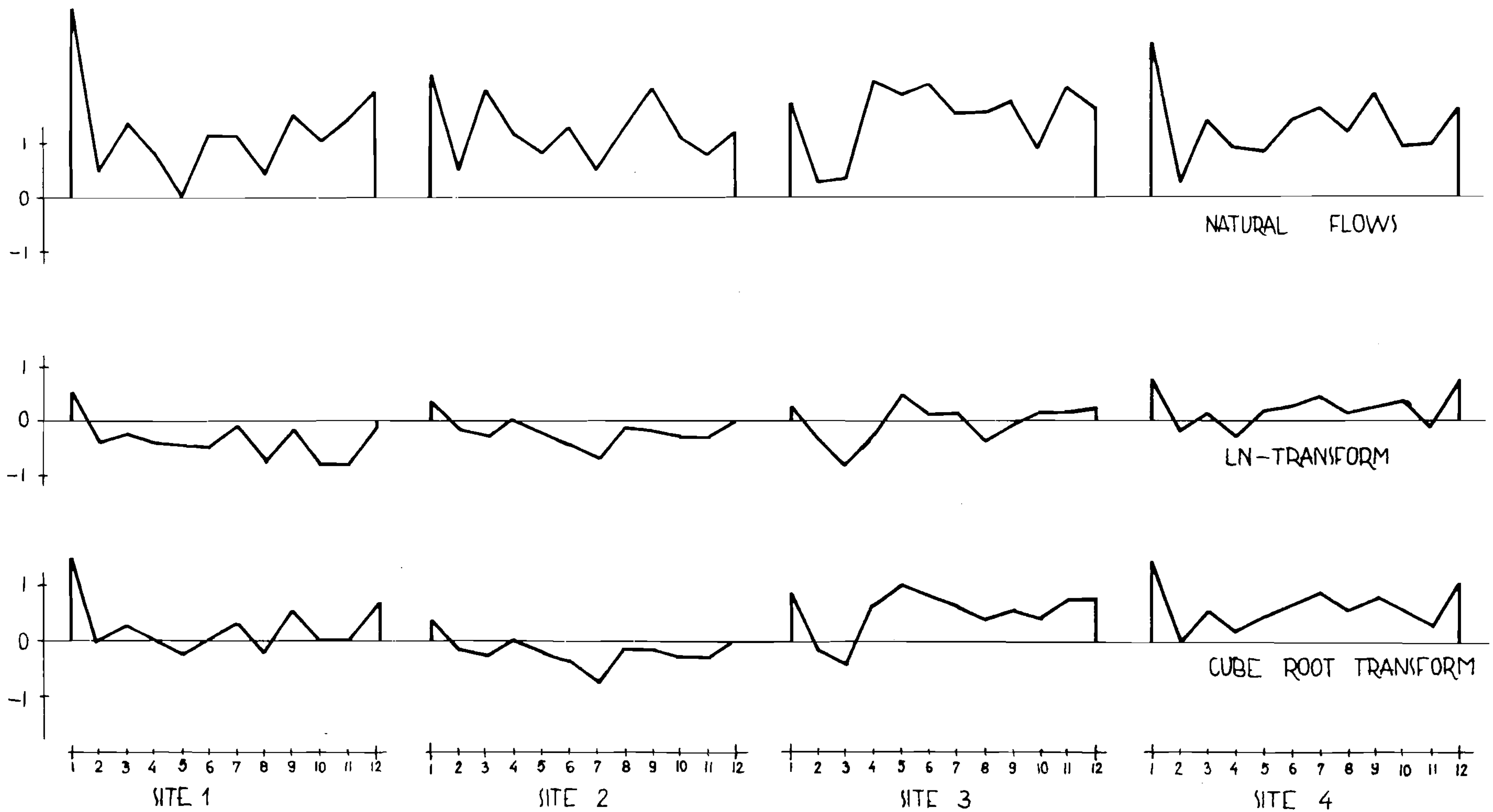
7. Comparison of statistical parameters of November flows (Czechoslovak Rivers).

8. Comparison of statistical parameters of December flows (Czechoslovak Rivers).

9. Comparison of flow frequency curves Site 3 - Czechoslovak Flow Data.

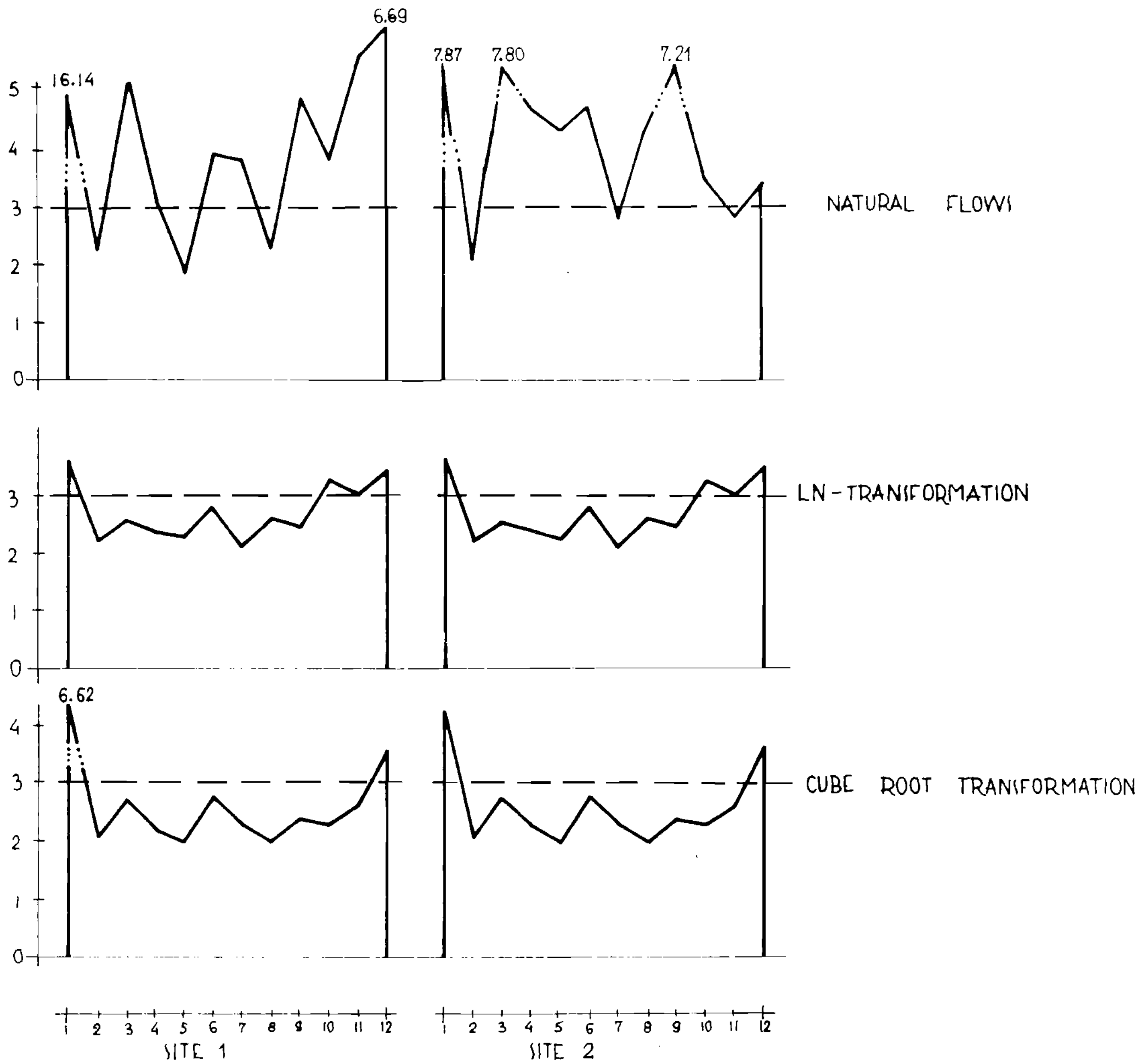
10. Comparison of cumulated frequency of reduced flow series at the level of 0.5-

11. Comparison of cumulated frequency of reduced flow series at the level of 1.0+



POLISH RIVERS — C_s VALUES OF NATURAL FLOW SEQUENCES AND SEQUENCES "NORMALIZED" BY LOGARITHMIC AND CUBE ROOT TRANSFORMATIONS

FIG. 1



POLISH RIVERS — KURTOSIS OF NATURAL FLOW SEQUENCES AND SEQUENCES "NORMALIZED" BY LOGARITHMIC AND CUBE ROOT TRANSFORMATIONS

FIG. 2

COMPARISON OF STATISTICAL PARAMETERS OF ANNUAL FLOW SEQUENCES
(POLISH RIVERS)

	S1			S2			S3			S4		
	H	G	A	H	G	A	H	G	A	H	G	A
MEAN	7.47	7.20	7.70	20.34	20.28	20.12	11.24	10.79	11.48	85.81	81.50	84.50
S.D.	2.16	2.48	3.51	5.22	6.34	7.25	3.40	3.86	4.08	19.22	19.78	21.13

1-2 INDICATES LAG-ZERO CROSS CORRELATION BETWEEN ANNUAL FLOWS AT SITES 1 AND 2, ETC.

1-1 INDICATES LAG-ONE SERIAL CORRELATION BETWEEN ANNUAL FLOWS AT SITE 1, ETC.

(1-2) INDICATES LAG-ONE CROSS CORRELATION BETWEEN ANNUAL FLOWS AT SITE 1 IN YEAR i AND ANNUAL FLOWS AT SITE 2 IN YEAR $i-1$, ETC.

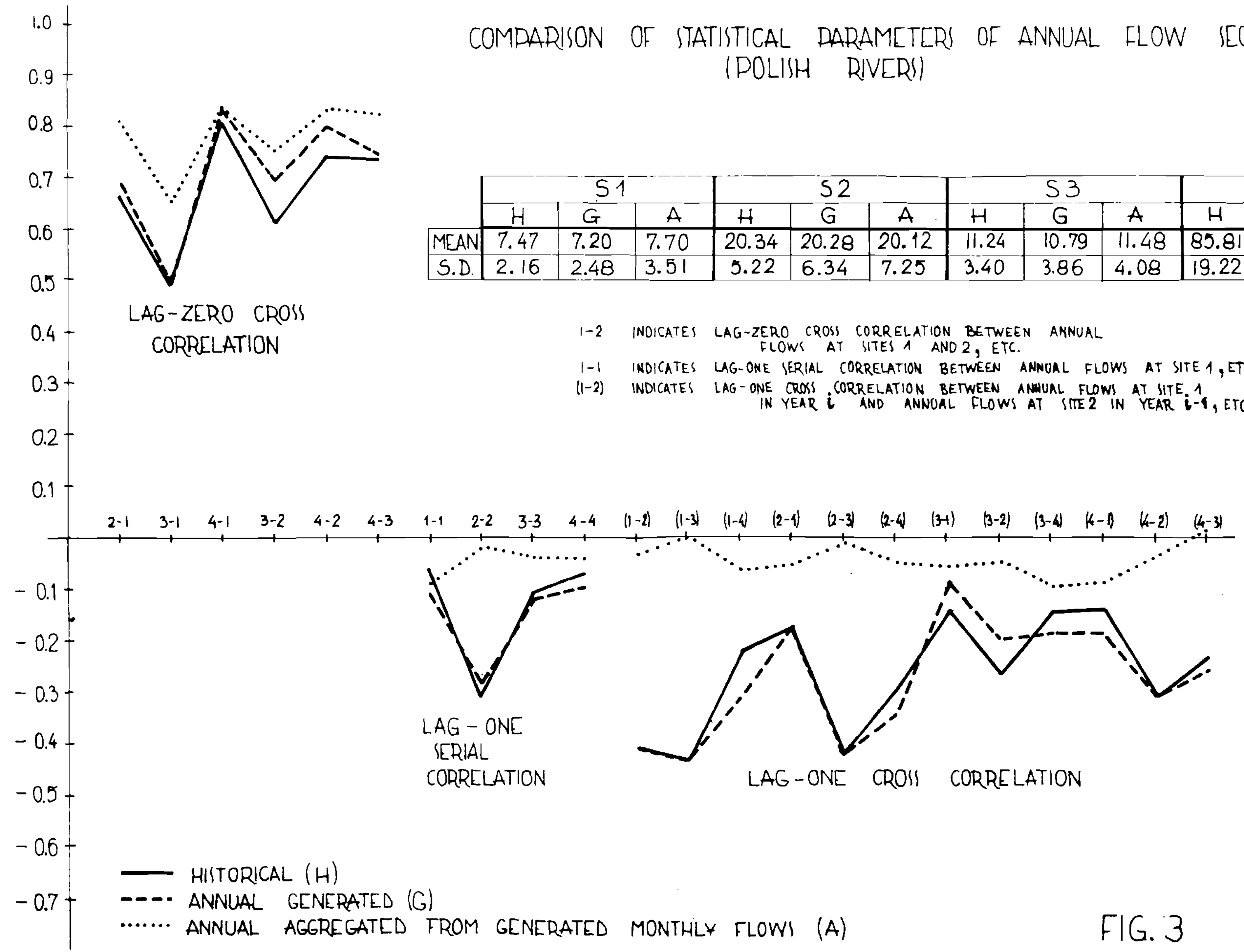
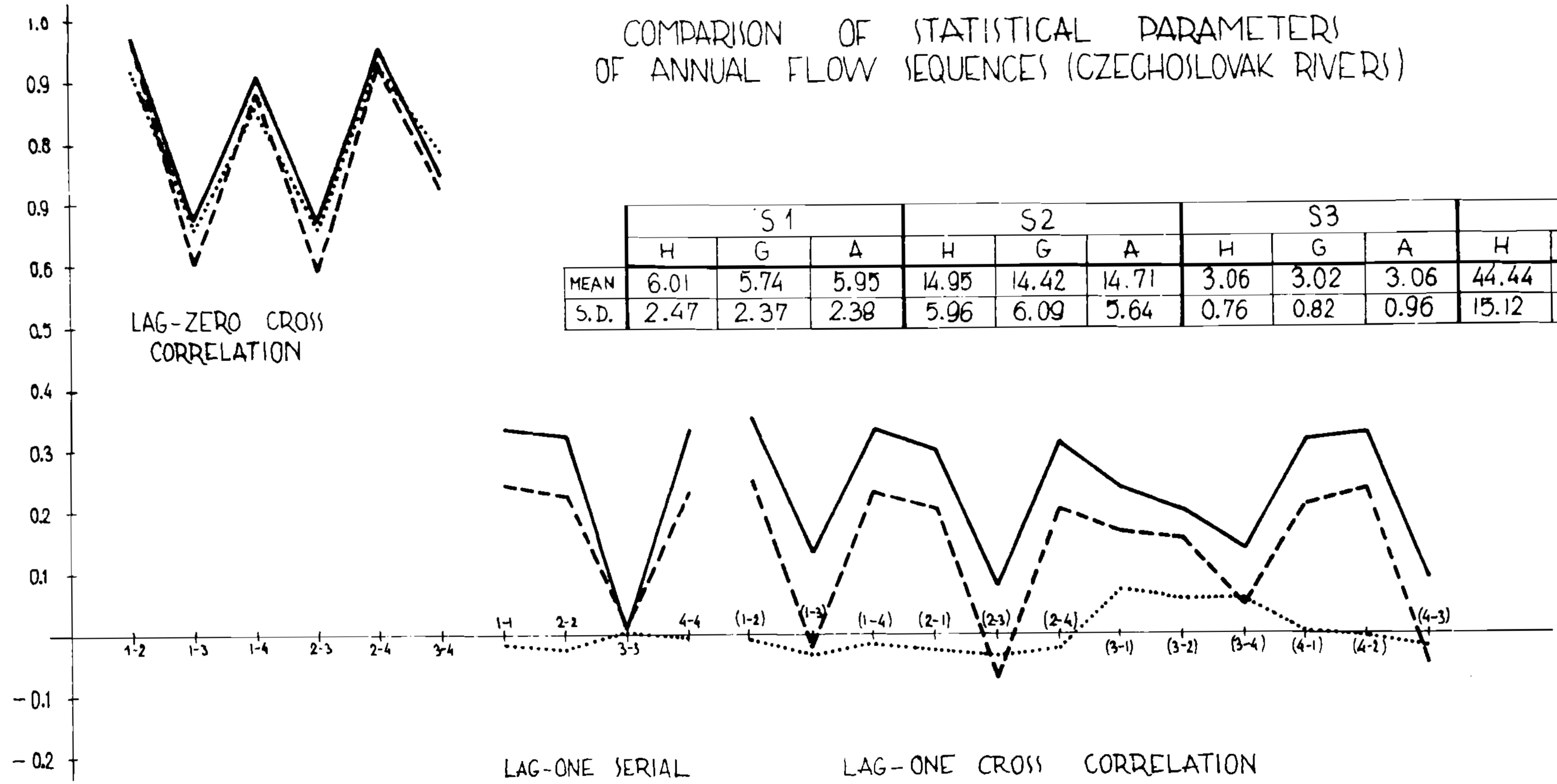


FIG. 3

COMPARISON OF STATISTICAL PARAMETERS OF ANNUAL FLOW SEQUENCES (CZECHOSLOVAK RIVERS)

	S1			S2			S3			S4		
	H	G	A	H	G	A	H	G	A	H	G	A
MEAN	6.01	5.74	5.95	14.95	14.42	14.71	3.06	3.02	3.06	44.44	42.89	43.79
S.D.	2.47	2.37	2.38	5.96	6.09	5.64	0.76	0.82	0.96	15.12	14.48	15.53



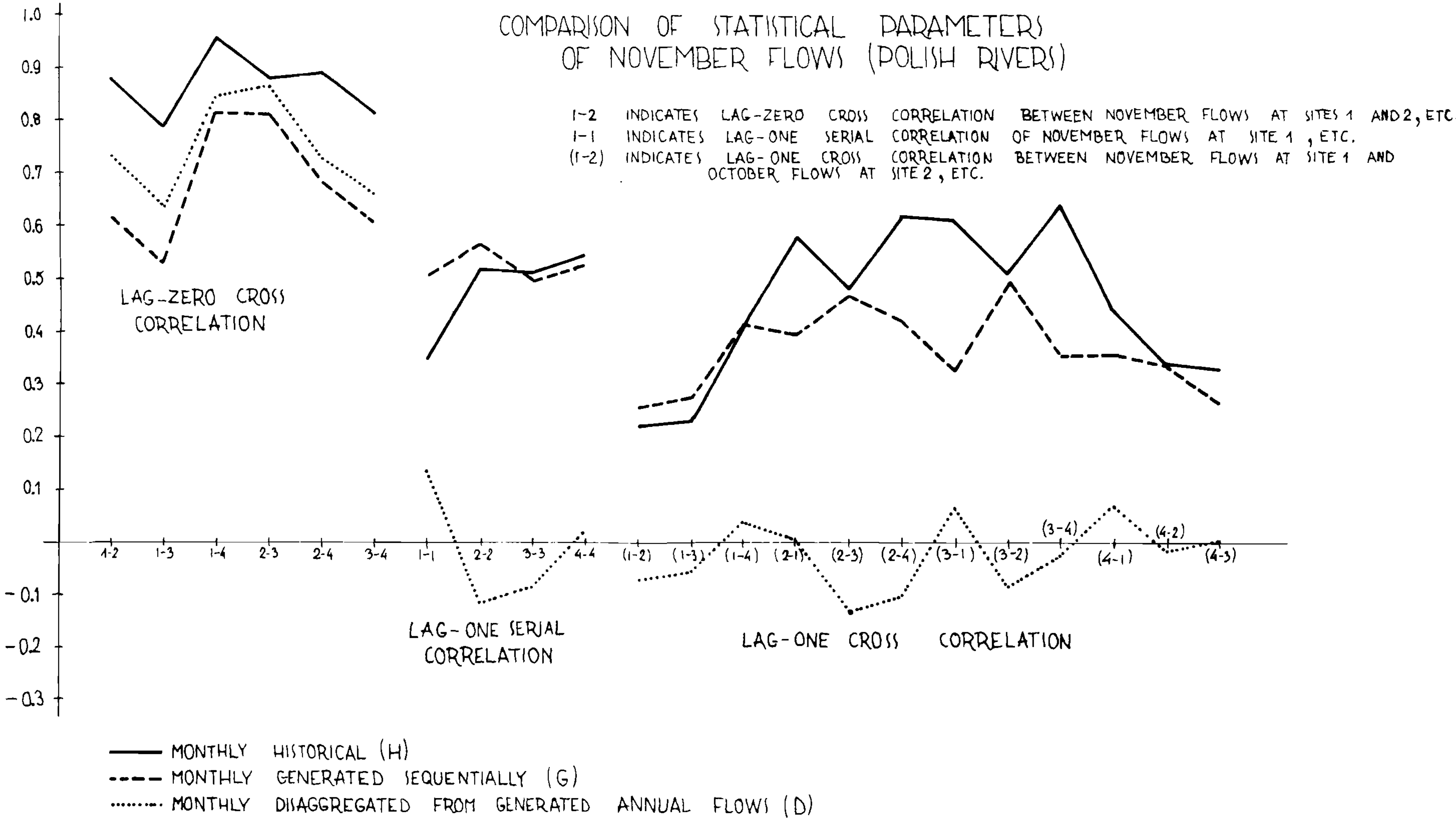
1-2 INDICATES LAG-ZERO CROSS CORRELATION BETWEEN ANNUAL FLOWS AT SITES 1 AND 2, ETC.

1-1 INDICATES LAG-ONE SERIAL CORRELATION BETWEEN ANNUAL FLOWS AT SITE 1, ETC.

(1-2) INDICATES LAG-ONE CROSS CORRELATION BETWEEN ANNUAL FLOWS AT SITE 1 IN YEAR i AND ANNUAL FLOWS AT SITE 2 IN YEAR $i-1$, ETC.

——— HISTORICAL (H)
- - - - ANNUAL GENERATED (G)
·········· ANNUAL AGGREGATED FROM GENERATED MONTHLY FLOWS (A)

FIG. 4

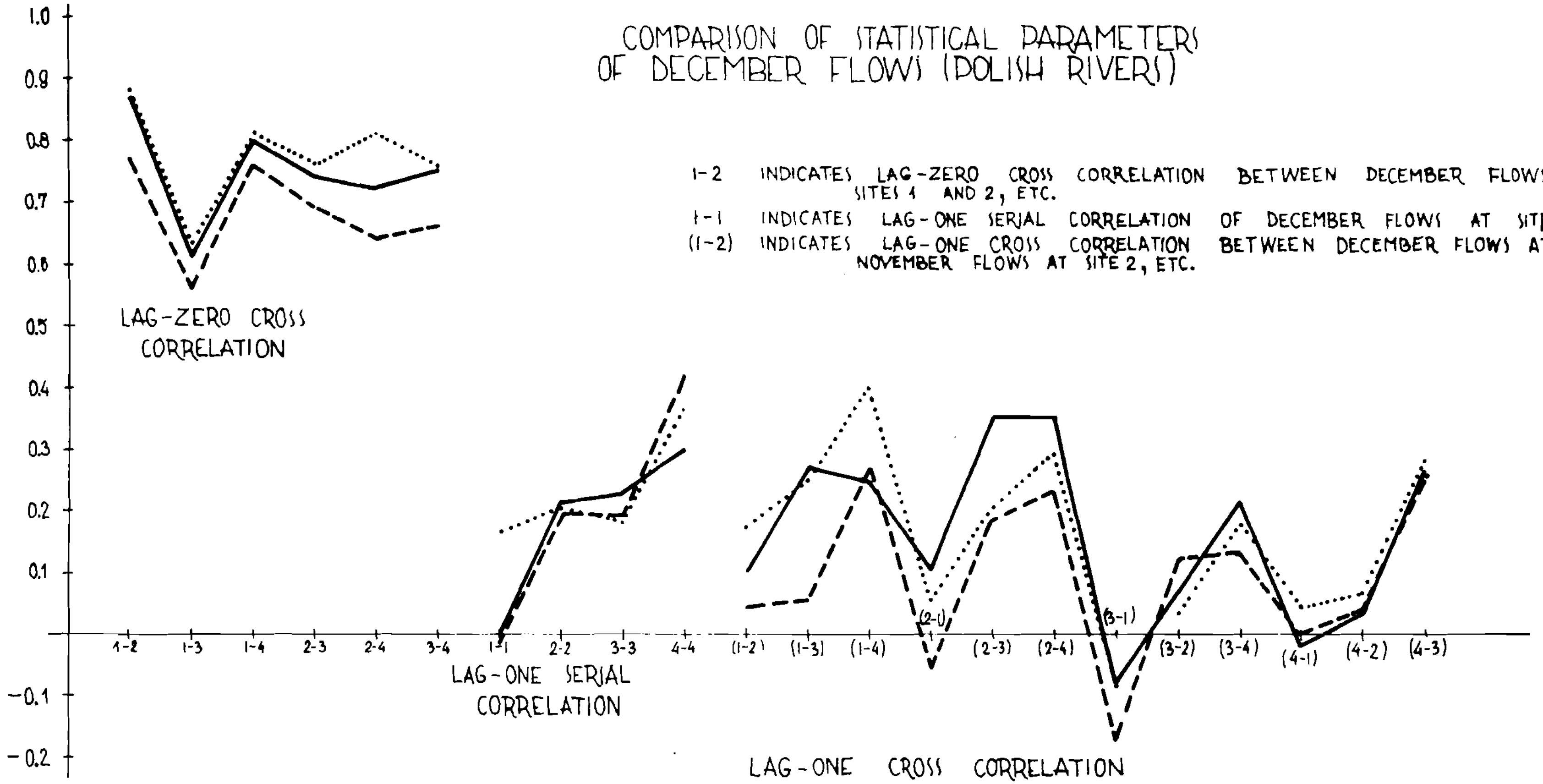


	S1			S2			S3			S4		
	H	G	D	H	G	D	H	G	D	H	G	D
MEAN	5.24	5.14	4.69	13.86	12.49	12.89	7.77	7.40	7.38	61.65	63.00	59.15
S.D.	6.20	5.88	3.74	12.95	12.11	9.43	6.04	6.54	5.58	39.76	33.82	26.08

FIG. 5

COMPARISON OF STATISTICAL PARAMETERS OF DECEMBER FLOWS (POLISH RIVERS)

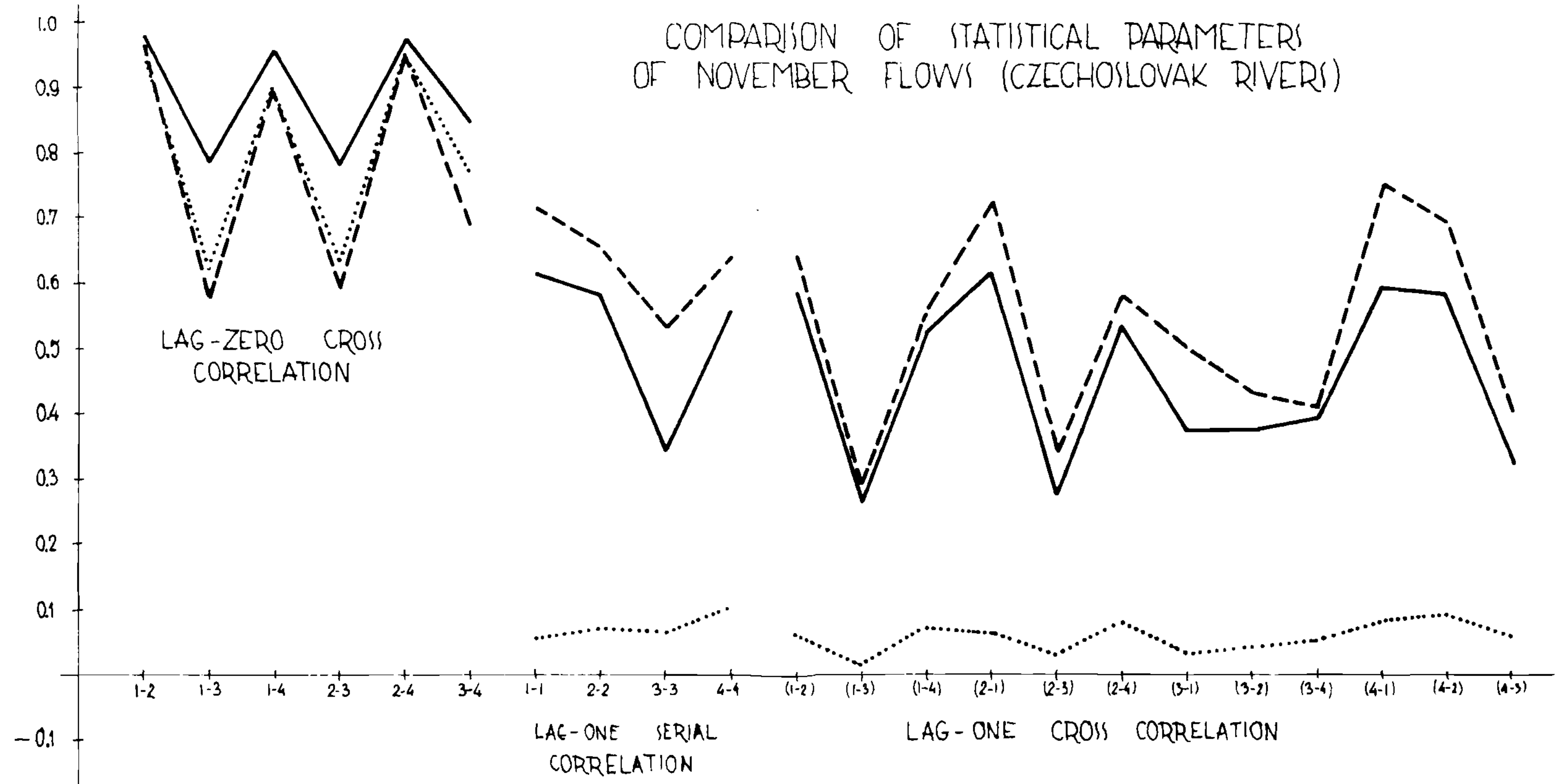
1-2 INDICATES LAG-ZERO CROSS CORRELATION BETWEEN DECEMBER FLOWS AT SITES 1 AND 2, ETC.
1-1 INDICATES LAG-ONE SERIAL CORRELATION OF DECEMBER FLOWS AT SITE 1, ETC.
(1-2) INDICATES LAG-ONE CROSS CORRELATION BETWEEN DECEMBER FLOWS AT SITE 1 AND NOVEMBER FLOWS AT SITE 2, ETC.



	S1			S2			S3			S4		
	H	G	D	H	G	D	H	G	D	H	G	D
MEAN	5.45	4.79	6.40	15.41	15.02	17.56	8.57	8.40	8.95	66.33	61.08	70.11
S.D	3.20	2.97	5.31	9.14	8.76	13.20	4.17	3.99	5.20	22.03	21.72	26.08

—— MONTHLY HISTORICAL (H)
- - - MONTHLY GENERATED SEQUENTIALLY (G)
······· MONTHLY DISAGGREGATED FROM GENERATED ANNUAL FLOWS (D)

FIG. 6

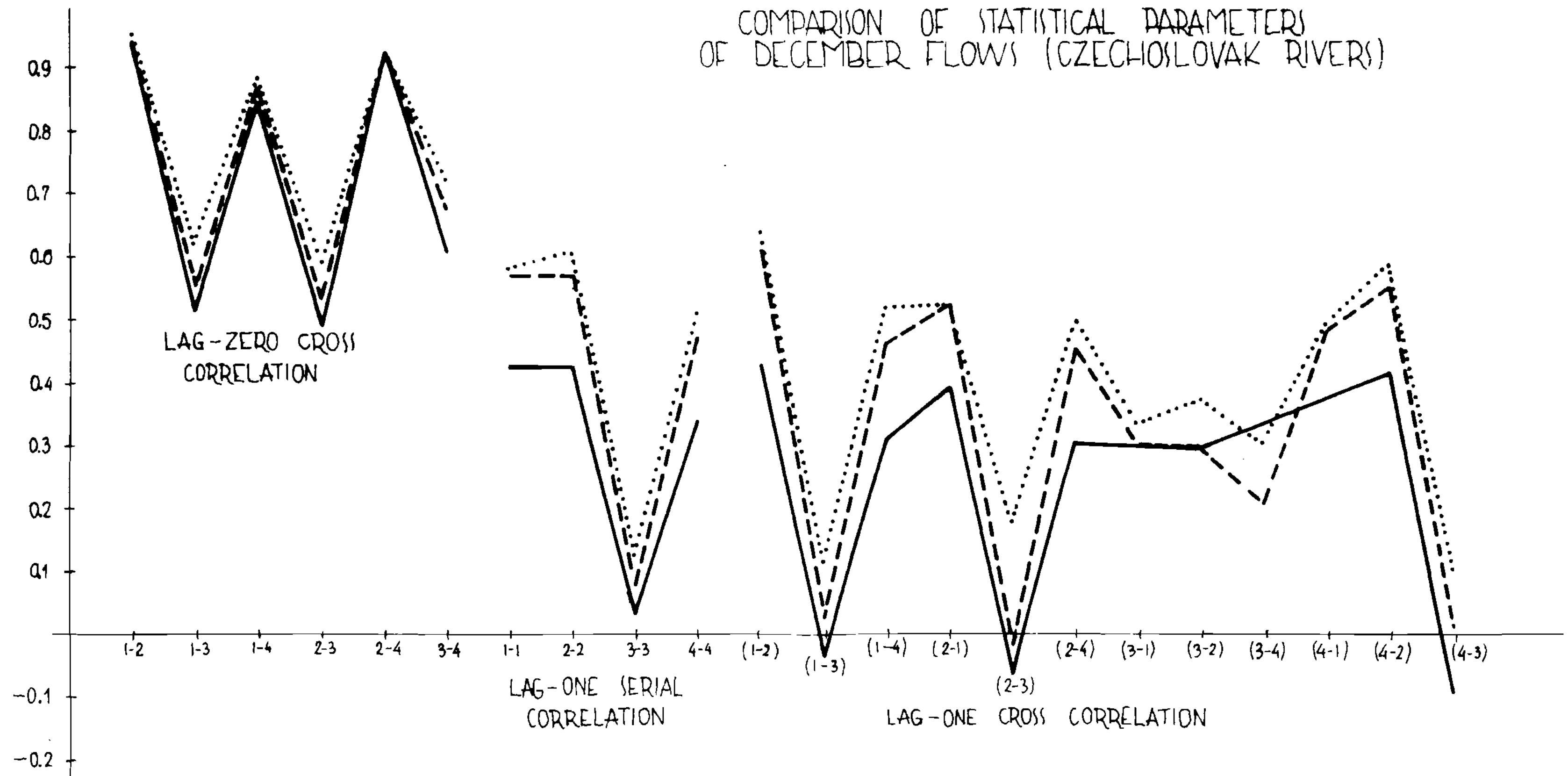


	S1			S2			S3			S4		
	H	G	D	H	G	D	H	G	D	H	G	D
MEAN	5.27	5.23	5.09	12.82	12.75	12.62	2.37	2.37	2.41	37.93	37.94	38.20
S.D	5.00	4.83	4.65	11.47	11.12	11.55	1.90	1.93	2.30	30.80	31.36	34.55

——— MONTHLY HISTORICAL (H)
- - - - MONTHLY GENERATED SEQUENTIALLY (G)
········· MONTHLY DISAGGREGATED FROM GENERATED ANNUAL FLOWS (D)

1-2 INDICATES LAG-ZERO CROSS CORRELATION BETWEEN NOVEMBER FLOWS AT SITES 1 AND 2, ETC.
1-1 INDICATES LAG-ONE SERIAL CORRELATION OF NOVEMBER FLOWS AT SITE 1, ETC.
(1-2) INDICATES LAG-ONE CROSS CORRELATION BETWEN NOVEMBER FLOWS AT SITE 1 AND OCTOBER FLOWS AT SITE 2, ETC.

FIG. 7



	S1			S2			S3			S4		
	H	G	D	H	G	D	H	G	D	H	G	D
MEAN	4.89	5.08	5.04	11.45	11.70	11.81	2.18	2.18	2.23	33.23	33.56	33.97
S.D.	3.87	3.89	3.54	8.58	8.27	8.09	1.22	1.48	1.41	19.85	19.62	20.15

——— MONTHLY HISTORICAL (H)
- - - - MONTHLY GENERATED SEQUENTIALLY (G)
········ MONTHLY DISAGGREGATED FROM GENERATED ANNUAL FLOWS (D)

1-2 INDICATES LAG-ZERO CROSS CORRELATION BETWEEN DECEMBER FLOWS AT SITES 1 AND 2, ETC.
1-1 INDICATES LAG-ONE SERIAL CORRELATION OF DECEMBER FLOWS AT SITE 1, ETC.
(1-2) INDICATES LAG-ONE CROSS CORRELATION BETWEEN DECEMBER FLOWS AT SITE 1 AND NOVEMBER FLOWS AT SITE 2, ETC.

FIG. 8

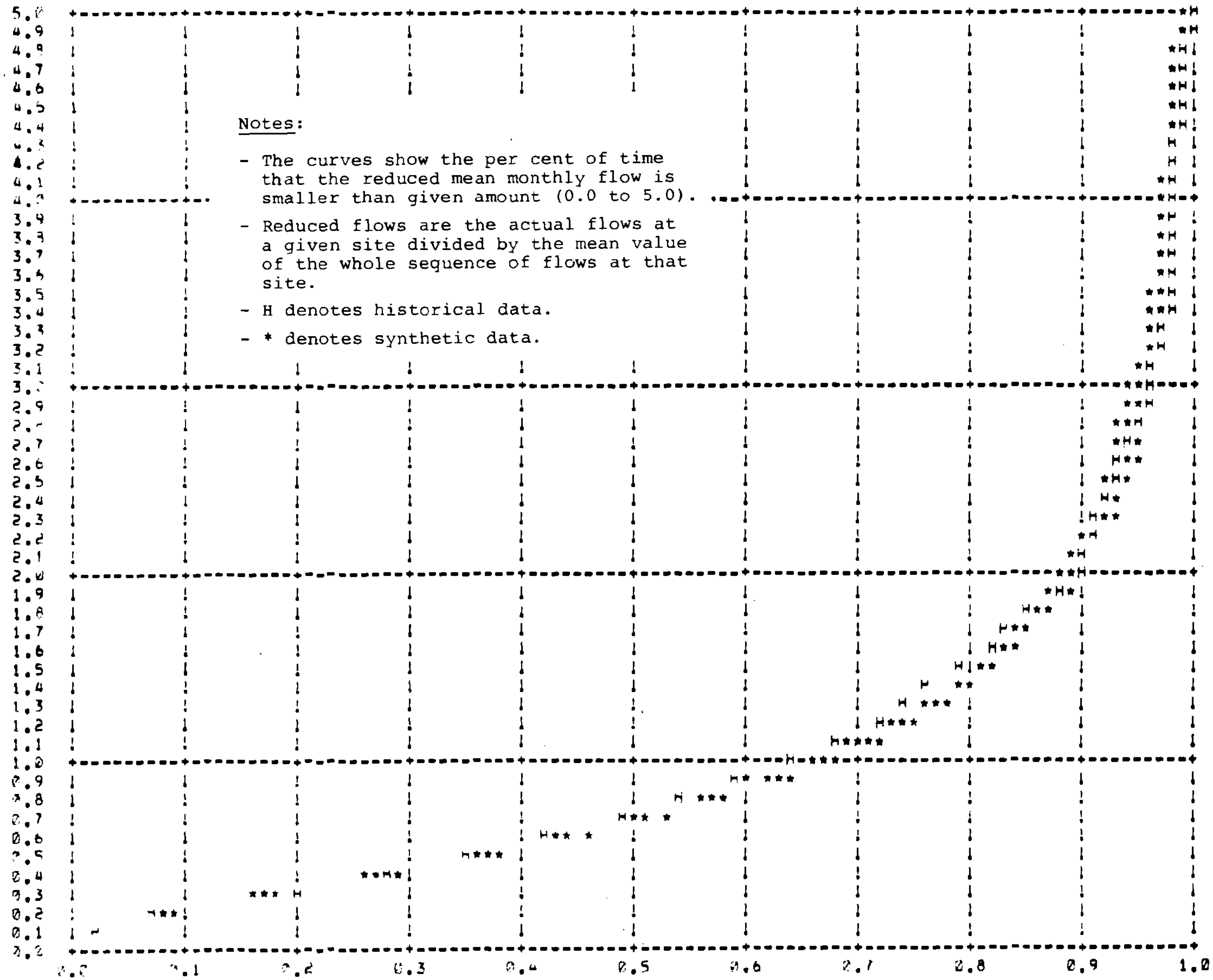
SITE:	S.T??					
FILE(S):	S.T	G.A	G.B	G.C	G.D	G.E
MEAN VALUES:	3.?6	3.?1	3.24	3.29	3.05	3.03

REDUCED FLOWS:

Figure 9

Comparison of flow frequency curves

Site 3 - Czechoslavak Flow Data



LEVEL: 0.5-

CUMULATED FREQUENCY VS SERIES LENGTH:

SITES: S.T01 S.T02 S.T03 S.T04

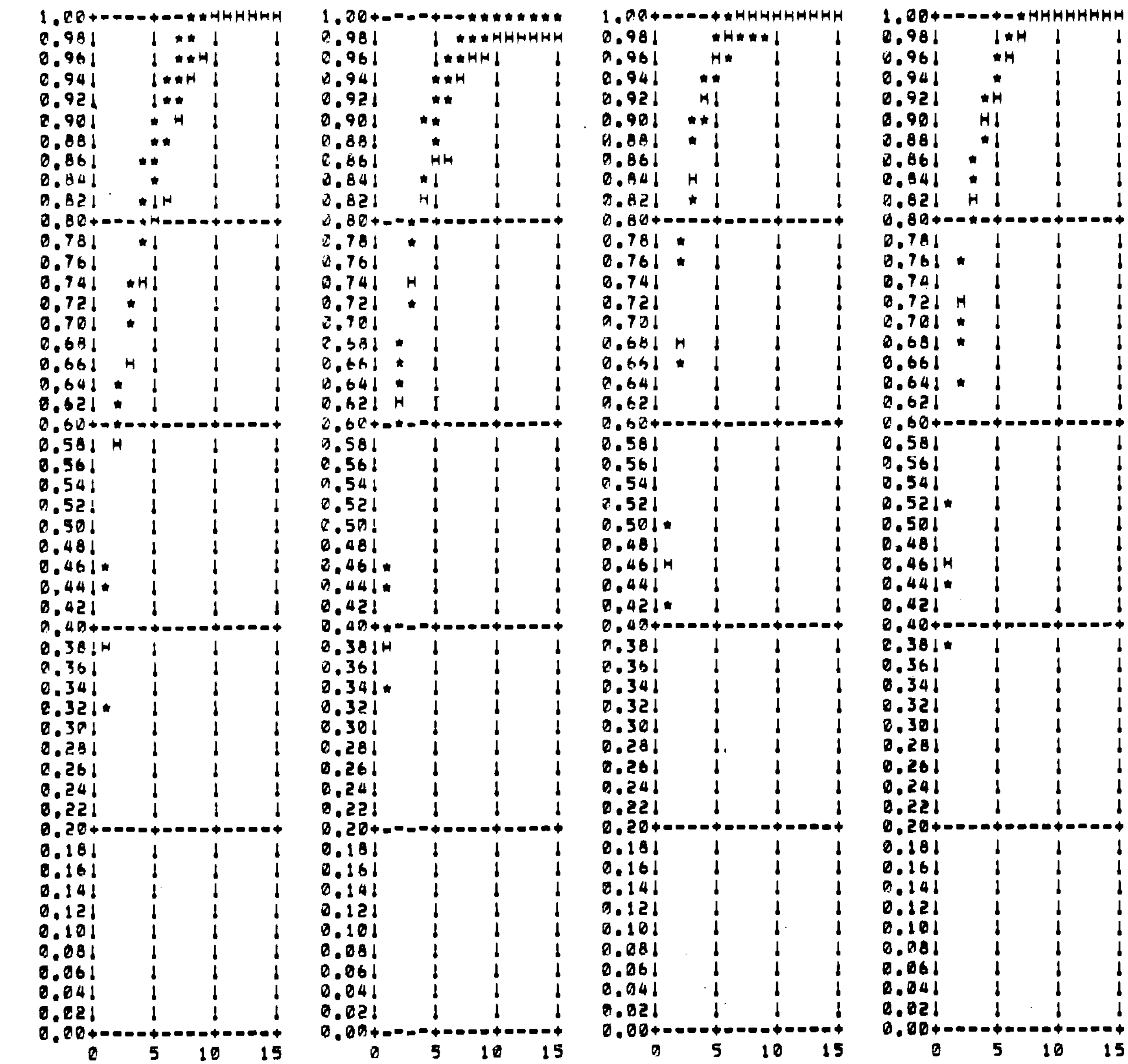


Figure 10

Comparison of cumulated frequency of reduced flow series at the level of 0.5-

Notes:

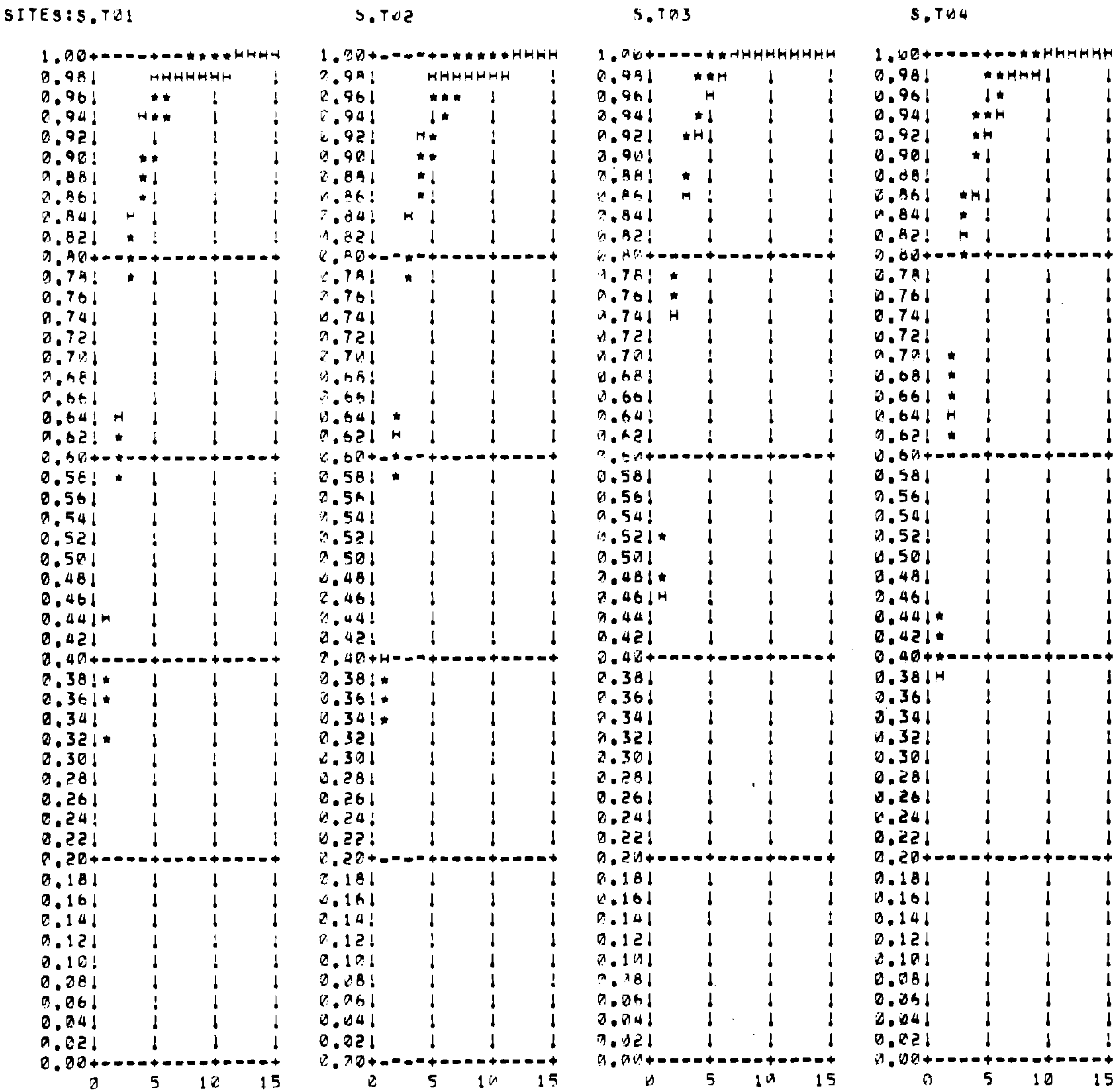
- The curves show the per cent of the total number of series of the duration (length) less or equal to the given value (1 to 15) at the level of 0.5-.
- H denotes historical data.
- * denotes synthetic data.

LEVEL: 1.0+

CUMULATED FREQUENCY VS SERIES LENGTH:

Figure 11

Comparison of cumulated frequency of reduced flow series at the level of 1.0+



Notes:

- The curves show the per cent of the total number of series of the duration (length) less or equal to the given value (1 to 15) at the level of 1.0+.
- H denotes historical data.
- * denotes synthetic data.

APPENDIX: COMPUTER IMPLEMENTATION

1. Introduction

The Multi-site Multi-season Streamflow Generation Package (MMSGP) is composed of the main segment, thirteen main subroutines and several auxiliary subroutines and functions. In addition, subroutines of the IBM Scientific Subroutine Package (MINV,EIGEN) are used. The generation of uniform distributed random numbers is accomplished by application of function RANDV - an assembly language function, that is available at PDP-11 computers, and is similar to the IBM RANDU function. The program is written in simplified FORTRAN, available under the UNIX system for PDP-11/45 computers. A listing of the main segment and all subroutines and functions (excluding IBM SSP subroutines) is provided at the end of the report.

MMSGP is designed as a collection of main subroutines, each intended for a small and independent task, rather than a program in the traditional sense, with fixed order of tasks. In other words, the program is 'parameter-driven' is such a way that the sequence of tasks for each execution of the program is determined by the sequence of input parameters or more precisely, groups of parameters, a group for each of the main subroutines. All the main subroutines are called from one main steering segment, which sequentially reads the groups of parameters in and selects the appropriate main subroutines. Such a structure enables a rather simple realization of various computational variants of flow generation without any modification of the package.

It is essential to mention that all data (historical and synthetic) are stored in the external (disk) files. The linkage of the main subroutines with appropriate data files is accomplished by the inclusion of the appropriate file names in the group of subroutine parameters. All file names are selected by the program user. The only values, which remain in the core memory are the generation or disaggregation parameters. Thus, the necessary core memory is kept as small as possible and does not depend on the length of historical and generated samples. However, it depends very much on the number of sites at which synthetic flows are generated and on the number of time seasons into which the year is divided. The present version of the package enables the generation of synthetic flows in up to five sites using not more than twelve time seasons (months).

2. <u>File Structures</u>

There are four different file structures:

1) basic historical data files,
2) lumped data files,
3) statistics files, and
4) matrices files.

<u>Basic historical data</u> files contain historical data records ordered according to the years sequence and arranged on a "one-file-one-site" principle. Each record contains two standard single fields which identify the year and the type of historical data (annual, quarterly or monthly flows). Moreover, it contains one single or multiple field representing the historical data. This is illustrated by the following table:

Indicator of data type	The meaning	Number of data	Length of record
1	annual	1	3
2	quarterly	4	6
3	monthly	12	14

Each of the basic data files may contain records with different data types, even for the same year. It is required, however, that the records are ordered according to the year sequence, and within each year they follow the order of data type indicators (from 1 to 3).

The creation and up-dating of basic historical data files are not covered by this description of the MMSGP.

<u>Lumped data files</u> contain the required data records for the selected sites. Each record forms a vector of values for the same year and the same period but for different sites. Records

are ordered on the time basis. The first record in the lumped data file is a description record, which contains the following fields:

1) file type (lumped data, statistics, matrices),
2) data type (historical, generated, disaggregated),
3) number of sites (ns),
4) number of seasons in a year (nt),
5) number of years (ny),
6) first historical year,
7) last historical year,
8) aggregation-disaggregation coefficient (nd), and
9) basic data file names (np to 5).

Statistics files contain seven records with the following characteristics:

1) description record,
2) mean values (ns * nt values),
3) standard deviation values (ns * nt values),
4) skewness coefficients (ns * nt values),
5) kurtosis coefficients (ns * nt values),
6) lag 0 correlation matrix (ns * ns * nt values), and
7) lag 1 correlation matrix (ns * ns * nt values).

Matrices files contain five records only, which are as follows:

1) description record,
2) mean values,
3) standard deviation values,
4) A/D - matrix, and
5) B/E - matrix.

3. General description of Input Parameters

The input parameters are divided into two types of the so-called main and secondary parameters. Main parameter records are distinguished by four leading asterisks in four leading character positions of the record. Secondary parameter records contain four space characters in these positions. All numerical parameters are integer numbers and should be right-justified in four-character fields. All text parameters should be left-justified and finished with space character. Each input parameter record contains one data only (number or text), which may be followed by a comment. Comments are not analysed by the program and may contain any character in any position. The input parameter records are composed into groups. The first record in each group has to be the main parameter record. The structure of groups is fixed and for each group depends on the main parameter record. The main parameter records may only contain 3-letter main subroutine name, optionally followed by 2-character trace indicator.

4. Main Segment

The main segment coordinates a set of sequentially called main subroutines and fixes the model of file operation tracing. The segment forms an interpretation loop, which in each cycle reads in the main parameter (being the name of the main subroutine) and a group of secondary parameters, depending of the main parameter. The cycle is finished by the calling sequence for indicated main subroutine. Thus the resulting sequence of main subroutines is determined by the set of input parameters for each program execution. "END" value of the main parameter is distinguished for termination of the program execution.

5. Main Subroutines

5.1 Subroutine SEL

The subroutine selects historical data records from indicated basic data fines and forms one lumped historical data file.

The parameter group is as follows:

```
****SEL
    NAME   OUTPUT-FILE-NAME
      FY   FIRST YEAR
      TY   LAST YEAR
      DI   DATA IDENT. (1,2 OR 3)
       N   NUMBER OF SITES (UP TO 5)
   NAME1   FIRST BASIC-FILE-NAME
   NAME2   SECOND         "
   ....
   NAMEN   LAST           "
```

5.2 Subroutine TRF

Subroutine TRF performs normalization and denormalization of data. This is done by transformation of one lumped file into another one.

The parameter group should be the following:

```
****TRF
     RT    REQUIRED TRANSFORMATION
   NAME1  INPUT-FILE-NAME
   NAME2  OUTPUT-FILE-NAME
```

where possible transformations include:

value of RT	transformation
0	no transformation
1	natural logarithm
-1	exponent
2	square root
-2	square
3	cube root
-3	cube

The input file for subroutine TRF has to be a lumped data file.

5.3 Subroutine MAM

Subroutine MAM computes the A and B matrices for streamflow generation (subroutine MAG), according to equations 2,3,4,5,7,8, 9, and 10.

The parameter group contains two records:

```
****MAM
   NAME   INPUT-FILE-NAME
```

Computed matrices remain in core memory.

5.4 Subroutine MYM

Subroutine MYM is similar to subroutine MAM, the computed A and B matrices, however, follow Young and Pisano (1967) standardization method.

The parameter group is similar as well:

```
****MYM
    NAME   INPUT-FILE-NAME
```

5.5 Subroutine MAG

Subroutine MAG uses the computed A and B matrices for generation of synthetic flows (equations 1 and 6), which are being written into an output lumped file.

The parameter group contains two numerical secondary parameters, the first of which determines the number of initial idle random number generator cycles, and the second, the number of years of synthetic data; the name of an output file is the third secondary parameter:

```
****MAG
       IG    NUMBER OF INITIAL GENERATIONS
       NY    NUMBER OF GENERATED YEARS
     NAME    OUTPUT-FILE-NAME
```

5.6 Subroutine DIM

This subroutine computes the D and E matrices for disaggregation (subroutine DIS), according to equations 12,13,14,15,17, 18,19,20, and 21.

The parameter group is as follows:

```
****DIM
    NAME1  HISTORICAL "X"-FILE-NAME
    NAME2  HISTORICAL "Y"-FILE-NAME
```

Computed matrices remain in core memory.

5.7 Subroutine DIS

Using the D and E matrices, stored in core memory, subroutine DIS performs disaggregation of synthetic input data flows, according to equations 11 and 16.

The parameter group contains two file names only:

```
****DIS
    NAME1  INPUT-FILE-NAME
    NAME2  OUTPUT-FILE-NAME
```

5.8 Subroutine RWC

Subroutine RWC enables storing into a file or restoring from a file computed statistics or generation and disaggregation matrices.

The parameter group takes the form:

```
****RWC
       OP   OPERATION: OP=1 READ, OP=2 WRITE
    NAME    FILE-NAME
```

5.9 Subroutine AGG

This subroutine computes mean values of indicated number of consecutive records of input lumped file, decreasing the number of seasons in the created output file. Such a process is called an aggregation of data. The aggregation coefficient has to be a divisor of actual (input) number of seasons.

The parameter group is as follows:

```
****AGG
       AC   AGGREGATION COEFFICIENT
    NAME1  INPUT-FILE-NAME
    NAME2  OUTPUT-FILE-NAME
```

5.10 Subroutine EST

Subroutine EST estimates the following statistics for the data recorded in the lumped input data file:

- mean values,
- standard deviation values,
- skewness coefficients,
- kurtosis coefficients,
- lag 0 correlation coefficients, and
- lag 1 correlation coefficients.

Computed statistics remain in core memory.

The parameter contains input file name only:

```
****EST
    NAME   INPUT-FILE-NAME
```

5.11 Subroutine OUT

Subroutine OUT prints two sets of computed statistics (subroutine EST) assuming that one set remains in core memory and the other one is stored in a file. Such an output form should make easier the comparison of appropriate values (for instance, the comparison of historical and synthetic statistics).

The parameter group contains one file name only:

```
****OUT
    NAME   STATISTICS-FILE-NAME
```

5.12 Subroutine FDA

This subroutine performs flow duration analysis and prints results in numerical and graphical forms. The analysis is being accomplished at one of six prefixed levels according to input parameter and can be used for up to 10 files simultaneously:

level-code:	value of reduced flow:
1	0.2
2	0.5
3	1.0
4	1.5
5	2.0
6	3.0

The parameter group is as follows:

```
****FDA
      L     LEVEL-CODE: -6≤L≤6, L≠0
      N     NUMBER OF FILES
  NAME1     FILE-NAME-1
  NAME2     FILE-NAME-2
  ....
```

5.13 Subroutine FFA

Subroutine FFA performs flow frequency analysis and prints results in two forms: numerical and graphical. The analysis can be used for up to 10 files at the same time.

The parameter group takes the form:

```
****FFA
      N     NUMBER OF FILES
  NAME1     FILE-NAME-1
  NAME2     FILE-NAME-2
  ....
```

6. Examples

6.1 Example 1: Generation

This example shows the sequence of parameters for the generation of monthly synthetic flows for the set of three sites. It is assumed that files SITE 1, SITE 2, and SITE 3 contain historical monthly data at least for the time period from 1931 until 1970.

After the selection of appropriate data and creation of the historical lumped data file SITE (subroutine SEL), historical statistics are estimated (subroutine EST) and stored in file STSITE (subroutine RWC). Subroutine TRF performs logarithmic transformation of historical data, creating the file SITELN, which is used for the computation of generation matrices (subroutine MAM). Subroutine MAG generates 100 years of synthetic monthly flows, stored in the file GENLN. Generated data are transformed (subroutine TRF) back to assumed lognormal distribution. Estimated statistics for synthetic data (subroutine EST) are printed together with historical statistics stored in the file STSITE.

```
****SEL
    SITE
    1931
    1970
       3
       3
    SITE1
    SITE2
    SITE3
****EST
    SITE
****RWC
       2
    STSITE
****TRF
       1
    SITE
    SITELN
****MAM
    SITELN
****MAG
       0
     100
    GENLN
****TRF
      -1
    GENLN
    GEN
****EST
    GEN
****OUT
    STSITE
****END
```

6.2 Example 2: Disaggregation

The files SITE, SITELN, and STSITE created in the previous example are used in the following sequence to show the disaggregation process.

The first four groups perform aggregation of historical monthly flows into historical quarterly and annual flows (files SITEQ and SITEA, respectively) and logarithmic transformation (files SITEQLN and SITEALN). Groups MAM and MAG accomplish the generation of synthetic annual flows (file GENALN). Two consecutive subsequences of subroutines DIM and DIS perform disaggregation of generated annual flows into synthetic quarterly flows (file GENQLN) and then quarterly flows into synthetic monthly ones (file GENLN). The final subsequence of subroutines TRF, EST, and OUT puts together statistics for historical and synthetic monthly data.

The application of log-transformation in the disaggregation process results in positive synthetic flows and follows the assumption of lognormal distribution of historical data in the Example 1. Such an assumption should be verified in some way, which is, however, beyond the scope of this section.

```
****AGG
       3
     SITE
     SITEQ
****AGG
       4
     SITEQ
     SITEA
****TRF
       1
     SITEQ
     SITEQLN
****TRF
       1
     SITEA
     SITEALN
****MAM
     SITEALN
****MAG
       0
      100
     GENALN
****DIM
     SITEALN
     SITEQLN
****DIS
     GENALN
     GENQLN
****DIM
     SITEQLN
     SITELN
****DIS
     GENQLN
     GENLN
****TRF
      -1
     GENLN
     GEN
****EST
     GEN
****OUT
     STSITE
****END
```

COMPUTER PROGRAM

```
MAY 28 09:12 1976   A:MAIN1.F PAGE 1

C       ******    M M S G P (V:1)    ******
C
C       INPUT PARAMETERS:
C        *)AGG:
C             AGGREGATION-FACTOR(I4)
C             INPUT-FILENAME(5A4)
C             OUTPUT-FILENAME(5A4)
C        *)DIM:
C             HIST-INPLIKE-FILENAME(5A4)
C             HIST-OUTLIKE-FILENAME(5A4)
C        *)DIS:
C             INPUT-FILENAME(5A4)
C             OUTPUT-FILENAME(5A4)
C        *)EST:
C             INPUT-FILENAME(5A4)
C        *)FDA:
C             LEVEL-CODE(I4)
C             NUMBER-OF-FILES(I4)
C             FILE-NAME-1(5A4)
C             ................
C        *)FFA:
C             NUMBER-OF-FILES(I4)
C             FILE-NAME-1(5A4)
C             ................
C        *)MAG:
C             NUMBER-OF-INITIAL-GENERATIONS(I4)
C             NUMBER-OF-GENERATED-YEARS(I4)
C             OUTPUT-FILENAME(5A4)
C        *)MAM:
C             INPUT-FILENAME(5A4)
C        *)MYM:
C             INPUT-FILENAME(5A4)
C        *)OUT:
C             STATISTICS-FILENAME(5A4)
C        *)RWC:
C             OPERATION-CODE(1-READ,2-WRITE)(I4)
C             INPUT/OUTPUT-FILENAME(5A4)
C        *)SEL:
C             OUTPUT-FILENAME(5A4)
C             YEAR-FROM(I4)
C             YEAR-TO(I4)
C             NUMBER-OF-PERIODS-CODE(I4)
C             NUMBER-OF-SITES(I4)
C             BASIC-FILENAME-1(5A4)
C              .............
C        *)TRF:
C             TRANSF-MODE(I4)
C             INPUT-FILENAME(5A4)
C             OUTPUT-FILENAME(5A4)
C
        DIMENSION A(2500)
        DIMENSION U(3),Z(14),II(8),F1(10),F2(5)
        COMMON/PARAM2/I1,I2,I3,I4,I5,I6,I7,I8,T(25)
        COMMON/WORK3/NF,FN(5,10)
        EQUIVALENCE (F1(6),F2(1)),(U(3),IU),(II(1),I1)
        DATA S/4H    /,G/4H****/,E1,E2/4H-    ,4H+    /
```

MAY 28 09:12 1976 A:MAIN1.F PAGE 2

```
      DATA Z(1),Z(2),Z(3),Z(4)/4HEND ,4HTRF ,4HEST ,4HMAM /
      DATA Z(5),Z(6),Z(7),Z(8)/4HMAG ,4HDIM ,4HDIS ,4HRWC /
      DATA Z(9),Z(10),Z(11),Z(12)/4HAGG ,4HOUT ,4HMYM ,4HSEL /
      DATA Z(13),Z(14)/4HFFA ,4HFDA /
      I1=0
      CALL FILEOP(0,U(1),1,0)
  200 CALL READ(3,U(1),1,G)
      N=0
      IF(U(2).EQ.E1) N=-1
      IF(U(2).EQ.E2) N=+1
      CALL FILEOP(0,U(1),1,N)
      MK=IU
      DO 210 N=1,14
      IF(U(1).EQ.Z(N)) GO TO 220
  210 CONTINUE
      CALL PRINT(26H UNDEFINED MAIN PARAMETER.,26,0,-1,0)
  001 STOP
  220 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14),N
  002 CALL READ(1,N,1,S)
      CALL READ(5,F1(1),2,S)
      CALL TRF(N,F1(1),F2(1),MK)
      GO TO 200
  003 CALL READ(5,F1(1),1,S)
      CALL EST(A(1),F1(1),0,MK)
      GO TO 200
  004 CALL READ(5,F1(1),1,S)
      CALL MAM(A(1),F1(1),0,MK)
      GO TO 200
  005 CALL READ(1,N,1,S)
      CALL READ(1,I5,1,S)
      CALL READ(5,F1(1),1,S)
      CALL MAG(A(1),N,F1(1),MK)
      GO TO 200
  006 CALL READ(5,F1(1),2,S)
      CALL DIM(A(1),F1(1),F2(1),MK)
      GO TO 200
  007 CALL READ(5,F1(1),2,S)
      CALL DIS(A(1),F1(1),F2(1),MK)
      GO TO 200
  008 CALL READ(1,N,1,S)
      CALL READ(5,F1(1),1,S)
      CALL RWC(N,A(1),F1(1),MK)
      GO TO 200
  009 CALL READ(1,N,1,S)
      CALL READ(5,F1(1),2,S)
      CALL AGG(N,F1(1),F2(1),MK)
      GO TO 200
  010 CALL READ(5,F1(1),1,S)
      CALL OUT(A(1),F1(1),MK)
      GO TO 200
  011 CALL READ(5,F1(1),1,S)
      CALL MYM(A(1),F1(1),0,MK)
      GO TO 200
  012 CALL READ(5,F1(1),1,S)
      CALL READ(1,I6,3,S)
      CALL READ(1,I3,1,S)
```

```
      CALL READ(5,T(1),I3,S)
      CALL SEL(F1(1),MK)
      GO TO 200
  013 CALL READ(1,NF,1,S)
      CALL READ(5,FN(1,1),NF,S)
      CALL FFA(A(1),MK)
      GO TO 200
  014 CALL READ(1,N,1,S)
      CALL READ(1,NF,1,S)
      CALL READ(5,FN(1,1),NF,S)
      CALL FDA(A(1),N,MK)
      GO TO 200
      END
```

```
MAY  3 14:43 1976  B:AGG.F PAGE 1

C       "AGG" - AGGREGATION OF DATA.
C
        SUBROUTINE AGG(NA,FI,FO,MK)
        DIMENSION FI(1),FO(1)
C
C       PARAMETERS:
C         NA - NUMBER OF AGGREGATED SEASONS,
C         FI - FILENAME OF ORIGINAL DATA,
C         FO - FILENAME OF AGGREGATED DATA,
C         MK - TRACE INDICATOR.
C
C       REMARK:
C         ORDERING OF DATA RECORDS:
C           YEAR.1  PERIOD.1  SITE.1
C                             SITE.2
C                             SITE.3
C                             ......
C                   PERIOD.2  SITE.1
C                             SITE.2
C           ......  ........  ......
C
C       REQUIRED:
C         FILEOP  PRINT   SETPAR
C
        COMMON/PARAM2/IDF,IDS,NS,NT,NY,IF,IT,MM,FN(5,5)
        COMMON/WORK1/R(60)
        COMMON/WORK2/Q(30)
        DIMENSION IA(8)
        EQUIVALENCE (IA(1),IDF)
        CALL FILEOP(1,FI(1),1,1)
        CALL SETPAR(1,1,IA(1),0,1)
        IF((NT/NA)*NA.NE.NT) CALL PRINT(4HAGG ,0,0.0,NA,-NT)
    200 MST=NT*NS
        NT=NT/NA
        NST=NT*NS
        FA=1.0/FLOAT(NA)
        CALL FILEOP(-1,FO(1),1,2)
        IF(IDS.GT.0) IDS=-IDS
        CALL SETPAR(2,2,IA(1),IDF,1)
        DO 300 I=1,NY
        CALL FILEOP(2,R(1),MST,1)
        DO 400 J=1,NS
        N=J
        DO 400 K=J,NST,NS
        S=0.0
        DO 600 L=1,NA
        S=S+R(N)
    600 N=N+NS
    400 Q(K)=S*FA
        IF(MK.NE.0) CALL PRINT(0,-1,Q(1),NS,NT)
    300 CALL FILEOP(3,Q(1),NST,2)
        CALL FILEOP(4,FI(1),1,1)
        CALL FILEOP(4,FO(1),1,2)
        RETURN
        END
```

MAY 7 09:31 1976 B:BAG.F PAGE 1

```
C       "BAG" - BAYESIAN GENERATION.
C
        SUBROUTINE BAG(A,NI,FN,MK)
        DIMENSION A(1),FN(1)
C
C      PARAMETERS:
C        A   - VECTOR OF GENERATION PARAMETERS:
C              MEAN(NS*NT)
C              SDEV(NS*NT)
C              "B"-MATRIX(NS*NS*NT)
C              "A"-MATRIX(NS*NS1*NT)
C              "XXT"-MATRIX(NS1*NS1*NT)
C        NI  - NUMBER OF INITIAL RANDOM NUMBER GENERATOR CYCLES.
C        FN  - OUTPUT-FILE-NAME.
C        MK  - TRACE INDICATOR.
C
C       REQUIRED:
C         COPY     FILEOP   MINV     MMULT    PRINT    RANDNV
C         SETPAR   SETPTR   VECOP    ZEROS
C
        COMMON/PARAM1/IA(8),FA(25)
        COMMON/PARAM2/IB(8),FB(25)
        COMMON/PARAM3/NS,NT,NY,NS1,NSS1,NST1,NSS,NST,NSS2,NSST1,NSST,LBB
        COMMON/WORK1/R(60)
        COMMON/WORK2/X(36)
        COMMON/WORK3/Z(6)
        COMMON/WORK4/Y(6)
        COMMON/WORK5/T(6)
        IF(IA(1).NE.5) CALL PRINT(4HBAG ,0,0,IA(1),-5)
        CALL SETPTR(IA(3),IA(4),IB(5),IA(3)+1)
        NSS2=NS1*NS1
        LBB=2*NST+1
        IF(NI.LE.0) GO TO 200
        DO 210 I=1,NI
  210   Q=RANDNV(1.0,0.0)
  200   CALL ZEROS(Z(2),NS)
        Z(1)=1.0
        S=1.0/FLOAT(IA(5)-NS)
        CALL COPY(IA(1),IB(1),33)
        IB(5)=NY
        IB(2)=4
        CALL FILEOP(1,FN(1),1,1)
        CALL SETPAR(1,2,IB(1),0,1)
        DO 300 I=1,NY
        LB=LBB
        LA=LB+NSST
        LX=LA+NSST1
        DO 310 J=1,NST,NS
        K=0
        DO 320 II=1,NS1
        Q=Z(II)
        DO 320 JJ=1,NS1
        K=K+1
        X(K)=A(LX)+Q*Z(JJ)
  320   LX=LX+1
        CALL MINV(X(1),NS1,Q,Y(1),T(1))
```

MAY 7 09:31 1976 B:BAG.F PAGE 2

```
      IF(ABS(Q).LT.1E-8) CALL PRINT(12H BAG: MINV ?,12,0,-1,0)
      Q=1.0
      K=0
      DO 330 II=1,NS1
      P=0.0
      DO 340 JJ=1,NS1
      K=K+1
  340 P=P+Z(JJ)*X(K)
  330 Q=Q-P*Z(II)
      IF(Q.LE.0.0) CALL PRINT(12H BAG: SQRT ?,12,0,-1,0)
      Q=SQRT(S/Q)
      DO 350 K=1,NS
  350 Y(K)=RANDNV(Q,0.0)
      CALL MMULT(A(LA),Z(1),T(1),NS,NS1,1)
      CALL MMULT(A(LB),Y(1),Z(2),NS,NS,1)
      CALL VECOP(Z(2),T(1),Z(2),NS,1)
      CALL COPY(Z(2),R(J),NS)
      LB=LB+NSS
  310 LA=LA+NSS1
      CALL VECOP(A,A(NST+1),R(1),NST,4)
      IF(MK.NE.0) CALL PRINT(0,-1,R(1),NS,NT)
  300 CALL FILEOP(3,R(1),NST,1)
      CALL FILEOP(4,FN(1),1,1)
      RETURN
      END
```

MAY 7 09:12 1976 BAM.F PAGE 1

```
C       "BAM" - ESTIMATION OF BAYESIAN MODEL PARAMETERS.
C
        SUBROUTINE BAM(A,F,MD,MK)
        DIMENSION A(1),F(1)
C
C       PARAMETERS:
C         A  - VECTOR OF PARAMETERS:
C                MEAN(NS*NT)
C                SDEV(NS*NT)
C                "B"-MATRIX(NS*NS*NT)
C                "A"-MATRIX(NS*NS1*NT)
C                "XXT"-MATRIX(NS1*NS1*NT)
C         F  - INPUT-FILE-NAME,
C         MD - MODE,
C         MK - TRACE INDICATOR.
C
C       REQUIRED:
C         COPY     COPYX    FILEOP   MINV     MMULT    PRINT
C         SETPAR   SETPTR   STAND    TRNG     VECOP    ZERUS
C
        COMMON/PARAM1/IA(8),T(5,5)
        COMMON/PARAM3/NS,NT,NY,NS1,NSS1,NST1,NSS,NST,NSS2,NSST1,NSST,NL
        COMMON/WORK1/R(65)
        COMMON/WORK2/Q(36)
        COMMON/WORK3/X(72)
        COMMON/WORK4/II(6)
        COMMON/WORK5/JJ(6)
        DIMENSION DD(2)
        DATA DD(1)/1.0/
        CALL STAND(A(1),F(1),IA(1),1)
        CALL FILEOP(1,F(1),1,1)
        CALL SETPAR(1,1,IA(1),0,1)
        CALL SETPTR(IA(3),IA(4),IA(5),IA(3)+1)
        NSS2=NS1*NS1
        NL=NT*(NSS+NSS1+NSS2)
        LBB=2*NST+1
        LAA=LBB+NSST
        LXX=LAA+NSST1
        CALL ZEROS(A(LBB),NL)
        CALL COPYX(DD(1),X(1),0,NS1,1,NT)
        I=NST+1-NS
        J=I+NST
        CALL VECOP(A(I),A(J),R(1),NS,3)
        DO 200 I=1,NY
        CALL FILEOP(2,R(NS1),NST,1)
        CALL VECOP(A(1),A(NST+1),R(NS1),NST,3)
        CALL COPYX(R(NS1),X(2),NS,NS1,NS,NT)
        CALL CORR(X(1),X(1),A(LXX),NS1,NS1,NT,0)
        CALL CORR(R(1),R(1),A(LBB),NS,NS,NT,0)
        CALL CORR(R(1),X(1),A(LAA),NS,NS1,NT,0)
  200   CALL COPY(R(NST+1),R(1),NS)
        CALL FILEOP(4,F(1),1,1)
        IF(MK.EQ.0) GO TO 210
        CALL PRINT(20H MATRIX: Y*TRANSP(Y),20,A(LBB),NSS,NT)
        CALL PRINT(20H MATRIX: X*TRANSP(Y),20,A(LAA),NSS1,NT)
        CALL PRINT(20H MATRIX: X*TRANSP(X),20,A(LXX),NSS2,NT)
```

```
  210 MT=NT
      IF(MD.EQ.0.OR.NT.EQ.1) GO TO 300
      DO 220 I=2,NT
      LB=LBB+(I-1)*NSS
      LA=LAA+(I-1)*NSS1
      LX=LXX+(I-1)*NSS2
      CALL VECOP(A(LB),A(LBB),A(LBB),NSS,1)
      CALL VECOP(A(LA),A(LAA),A(LAA),NSS1,1)
  220 CALL VECOP(A(LX),A(LXX),A(LXX),NSS2,1)
      DD(2)=1.0/FLOAT(NT)
      CALL VECOP(DD(2),A(LBB),A(LBB),NSS,5)
      CALL VECOP(DD(2),A(LAA),A(LAA),NSS1,5)
      CALL VECOP(DD(2),A(LXX),A(LXX),NSS2,5)
      MT=1
  300 DO 310 I=1,MT
      LB=LBB+(I-1)*NSS
      LA=LAA+(I-1)*NSS1
      LX=LXX+(I-1)*NSS2
      CALL COPY(A(LX),Q(1),NSS2)
      CALL MINV(Q(1),NS1,D,II(1),JJ(1))
      IF(ABS(D).LT.1E-8) CALL PRINT(12H BAM: MINV ?,12,0,-1,0)
      CALL COPY(A(LA),X(1),NSS1)
      CALL MMULT(X(1),Q(1),A(LA),-NS,NS1,NS1)
      CALL MMULT(A(LA),X(1),Q(1),NS,NS1,NS1)
      CALL VECOP(A(LB),Q(1),Q(1),NSS,2)
  310 CALL TRNG(Q(1),A(LB),NS)
      IF(MD.EQ.0.OR.NT.EQ.1) GO TO 320
      DO 330 I=2,NT
      LB=LBB+(I-1)*NSS
      LA=LAA+(I-1)*NSS1
      LX=LXX+(I-1)*NSS2
      CALL COPY(A(LBB),A(LB),NSS)
      CALL COPY(A(LAA),A(LA),NSS1)
  330 CALL COPY(A(LXX),A(LX),NSS2)
  320 IF(MK.EQ.0) GO TO 340
      CALL PRINT(12H MATRIX: A  ,12,A(LAA),NSS1,NT)
      CALL PRINT(12H MATRIX: B  ,12,A(LBB),NSS,NT)
  340 IA(1)=5
      RETURN
      END
```

```
MAY  3 16:23 1976  H:DIM.F PAGE 1


C      "DIM" - ESTIMATION OF DISAGGREGATION PARAMETERS.
C
       SUBROUTINE DIM(A,F1,F2,MK)
       DIMENSION A(1),F1(1),F2(1)
C
C      PARAMETERS:
C        A  - VECTOR OF DISAGGREGATION PARAMETERS:
C             MEAN(NSD*NT)
C             SDEV(NSD*NT)
C             YX(NSD*NS*NT),    D(NSD*NS*NT)
C             YY(NSD*NSD*NT),   E(NSD*NSD*NT)
C             XX(NS*NS*NT)
C        F1 - NAME OF HISTORICAL, INP-LIKE FILE,
C        F2 - NAME OF HISTORICAL, OUT-LIKE FILE,
C        MK - TRACE INDICATOR.
C
C      REQUIRED:
C        CMAB    COPY    CORR    FILEOP  PRINT   SETPAR  SETPTR
C        STAND   VECOP   ZEROS
C
       COMMON/PARAM1/IA(8),FNA(25)
       COMMON/PARAM2/IB(8),FNB(25)
       COMMON/PARAM3/NS,NT,NY,ND,NSD,NTD,NSS,NST,NSSD,NSTD,NSST,NSSDD
       COMMON/WORK1/R2(60)
       COMMON/WORK2/R1(30)
       COMMON/WORK3/S(100)
       DIMENSION FF(1)
       EQUIVALENCE (FF(1),FY)
       CALL STAND(A(1),F2(1),IA(1),0)
       CALL STAND(S(1),F1(1),IB(1),0)
       CALL FILEOP(1,F1(1),1,1)
       CALL FILEOP(1,F2(1),1,2)
       CALL SETPAR(2,1,IA(1),0,1)
       CALL SETPAR(1,1,IB(1),0,3)
       CALL SETPTR(IA(3),IB(4),IA(5),IA(4)/IB(4))
       IF(ND*NT.NE.IA(4)) CALL PRINT(4HDAB ,0,0.0,IA(4),-NT)
       LYX=NSTD+NSTD+1
       LYY=LYX+NSSD*NT
       LXX=LYY+NSSDD*NT
       LRR=LXX+NSST
       LD=(1+ND+ND*ND)*NSST
C
C        CALCULATE XX, YX, YY MATRICES:
C
       CALL ZEROS(A(LYX),LD)
       DO 200 I=1,NY
       CALL FILEOP(2,R1(1),NST,1)
       CALL FILEOP(2,R2(1),NSTD,2)
       CALL VECOP(S(1),S(NST+1),R1(1),NST,3)
       CALL VECOP(A(1),A(NSTD+1),R2(1),NSTD,3)
       CALL CORR(R1(1),R1(1),A(LXX),NS,NS,NT,0)
       CALL CORR(R1(1),R2(1),A(LYX),NS,NSD,NT,0)
  200  CALL CORR(R2(1),R2(1),A(LYY),NSD,NSD,NT,0)
       CALL FILEOP(4,F1(1),1,1)
       CALL FILEOP(4,F2(1),1,2)
       FY=1.0/FLOAT(NY)
```

MAY 3 16:23 1976   B:DIM.F PAGE 2

```
      CALL CORR(FF(1),A(LYX),A(LYX),LD,1,1,1)
C
C        CALCULATE D, E MATRICES:
C
      DO 300 I=1,NT
      IF(MK.EQ.0) GO TO 310
      CALL PRINT(4H  T:,0,0,I,0)
      CALL PRINT(12H MATRIX: XX ,12,A(LXX),NS,NS)
      CALL PRINT(16H MATRIX: (YX)I   ,16,A(LYX),-NS,NSD)
      CALL PRINT(12H MATRIX: YY ,12,A(LYY),NSD,NSD)
  310 CALL COPY(A(LYX),S(1),NSSD)
      CALL COPY(A(LXX),A(LRR),NSS)
      CALL CMAB(A(LRR),A(LYY),A(LYX),S(1),NS,NSD)
      IF(MK.EQ.0) GO TO 320
      CALL PRINT(12H MATRIX:(D)T,12,A(LYX),-NS,NSD)
      CALL PRINT(12H MATRIX: E  ,12,A(LYY),NSD,NSD)
  320 LYY=LYY+NSSDD
      LYX=LYX+NSSD
  300 LXX=LXX+NSS
      IA(1)=4
      IA(8)=ND
      RETURN
      END
```

```
MAY  4 09:21 1976  B:DIS.F PAGE 1

C       "DIS" - DISAGGREGATION OF SYNTHETIC DATA.
C
        SUBROUTINE DIS(A,FI,FO,MK)
        DIMENSION A(1),FI(1),FO(1)
C
C       PARAMETERS:
C          A  - VECTOR OF DISAGGREGATION PARAMETERS:
C                MEAN(NS*NT*ND)
C                SDEV(NS*NT*ND)
C                D(NSD*NS*NT)
C                E(NSD*NSD*NT)
C         FI - INPUT-FILE-NAME,
C         FO - OUTPUT-FILE-NAME,
C         MK - TRACE INDICATOR.
C
C       REQUIRED:
C         FGEN     FILEOP  PRINT   SETPAR  SETPTR  STAND   VECOP
C
        COMMON/PARAM1/IA(8),FN1(25)
        COMMON/PARAM2/IB(8),FN2(25)
        COMMON/PARAM3/NS,NT,NY,ND,NSD,NTD,NSS,NST,NSSD,NSTD,NSST,NSSDD
        COMMON/WORK1/R(60)
        COMMON/WORK2/Q(30)
        COMMON/WORK4/S(30)
        IF(IA(1).NE.4) CALL PRINT(4HDIS ,0,0.0,IA(1),-4)
        CALL STAND(S(1),FI(1),IB(1),0)
        CALL FILEOP(1,FI(1),1,1)
        CALL SETPAR(1,1,IB(1),0,6)
        CALL SETPTR(IA(3),IB(4),IB(5),IA(8))
        CALL FILEOP(-1,FO(1),1,2)
        IB(4)=NTD
        IB(2)=3
        CALL SETPAR(2,2,IB(1),0,1)
        NA=NSTD+NSTD+1
C
C         DISAGGREGATE:
C
        DO 200 I=1,NY
        NAA=NA
        NBB=NA+NSST*ND
        N=1
        CALL FILEOP(2,Q(1),NST,1)
        CALL VECOP(S(1),S(NST+1),Q(1),NST,3)
        DO 210 J=1,NST,NS
        CALL FGEN(A(NAA),A(NBB),Q(J),R(N),NS,NSD)
        N=N+NSD
        NAA=NAA+NSSD
  210   NBB=NBB+NSSDD
        CALL VECOP(A(1),A(NSTD+1),R(1),NSTD,4)
        IF(MK.NE.0) CALL PRINT(0,-1,R(1),NS,NTD)
  200   CALL FILEOP(3,R(1),NSTD,2)
        CALL FILEOP(4,FI(1),1,1)
        CALL FILEOP(4,FO(1),1,2)
        RETURN
        END
```

```
APR 30 12:22 1976  B:EST.F PAGE 1

C       "EST" - ESTIMATION OF STATISTICS.
C
        SUBROUTINE EST(A,F,M,MK)
        DIMENSION A(1),F(1)
C
C       PARAMETERS:
C         A   - VECTOR OF STATISTICAL PARAMETERS:
C                 MEAN(NS*NT)
C                 SDEV(NS*NT)
C                 SKEW(NS*NT)
C                 KURT(NS*NT)
C                 CORR-LAG-0(NS*NS*NT)
C                 CORR-LAG-1(NS*NS*NT)
C         F   - FILENAME OF INPUT DATA,
C         M   - MODE,
C         MK  - TRACE INDICATOR.
C
C       REQUIRED:
C         ECORR    FILEOP   MNSD     SETPAR   SETPTR   STAND
C         VECOP    ZEROS
C
        COMMON/PARAM1/IA(8),FNA(25)
        COMMON/PARAM3/NS,NT,NY,NK,NSK,NTK,NSS,NST,NSSK,NSTK,M1,M2
        COMMON/WORK1/R(60)
        DIMENSION FF(1)
        EQUIVALENCE (FF(1),FY)
        CALL STAND(A(1),F(1),IA(1),1)
        CALL ECORR(A(1),A(4*NST+1),F(1),IA(1),1,M)
        CALL FILEOP(1,F(1),1,1)
        CALL SETPAR(1,1,IA(1),0,1)
        CALL SETPTR(IA(3),IA(4),IA(5),2)
        M1=NST+1
        M2=NSTK+1
        FY=1.0/FLOAT(NY)
        CALL ZEROS(A(M2),NSTK)
        DO 100 I=1,NY
        CALL FILEOP(2,R(1),NST,1)
        CALL VECOP(A(1),A(M1),R(1),NST,3)
    100 CALL MNSD(R(1),A(M2),NST,-1)
        CALL FILEOP(4,F(1),1,1)
        CALL VECOP(FF(1),A(M2),A(M2),NSTK,5)
        IF(MK.EQ.0) GO TO 800
        CALL PRINT(8H  MEAN: ,8,A(1),NS,NT)
        CALL PRINT(8H  SDEV: ,8,A(M1),NS,NT)
        CALL PRINT(8H  SKEW: ,8,A(M2),NS,NT)
        CALL PRINT(8H  KURT: ,8,A(3*NST+1),NS,NT)
        M1=4*NST+1
        M2=M1+NSS*NT
        CALL PRINT(12H CORR-LAG-0:,12,A(M1),NSS,NT)
        CALL PRINT(12H CORR-LAG-1:,12,A(M2),NSS,NT)
    800 IA(1)=2
        IA(8)=1
        RETURN
        END
```

JUL 27 10:57 1976 B:FDA.F PAGE 1

```
C      "FDA" - FLOW DURATION ANALYSIS.
C
       SUBROUTINE FDA(A,IN,MK)
       DIMENSION A(15,5,10)
C      PARAMETERS:
C        A  - WORKING AREA,
C        IN - LEVEL-CODE,
C        MK - TRACE INDICATOR.
C
C      REQUIRED:
C        FILEOP  SETPAR  STAND1  ZEROS
C
       COMMON/PARAM1/IZ(1),MM,NS,NT,NY,NST,L,K,FA(5,5)
       COMMON/WORK1/T(80)
       COMMON/WORK2/EV(5),TT(5),LL(5),MT(10)
       COMMON/WORK3/NF,F(5,10)
       DIMENSION R(16,5),Z(6)
       EQUIVALENCE (T(1),R(1,1))
       DATA Z(1),Z(2),Z(3),Z(4),Z(5),Z(6)/0.2,0.5,1.0,1.5,2.0,3.0/
       DATA S1,S2,S3,S4,X1,X2/4H+   ,4H-   ,4HI   ,4H    ,4H*   ,4HH   /
       IA=IABS(IN)
       XX=S1
       IF(IA.NE.IN) XX=S2
       WRITE(6,100)Z(IA),XX
       II=1
       DO 200 I=1,NF
       CALL STAND1(EV(1),F(1,I),IZ(1),II)
       CALL FILEOP(1,F(1,I),1,1)
       CALL SETPAR(1,1,IZ(1),0,II)
       MT(I)=MM
       NST=NS*NT
       CALL ZEROS(A(1,1,I),15*NS)
       CALL ZEROS(TT(1),NS)
       DO 210 J=1,NS
  210  LL(J)=0
       DO 300 J=1,NY
       CALL FILEOP(2,T(1),NST,1)
       DO 300 M=1,NS
       YY=Z(IA)*EV(M)
       DO 300 N=M,NST,NS
       IF(IN) 361,362,363
  362  STOP
  361  IF(T(N).GE.YY) GO TO 370
  380  LL(M)=LL(M)+1
       GO TO 300
  363  IF(T(N).GE.YY) GO TO 380
  370  IF(LL(M).EQ.0) GO TO 300
       TT(M)=TT(M)+1.0
       IF(LL(M).LE.15) A(L,M,I)=A(L,M,I)+1.0
       LL(M)=0
  300  CONTINUE
       DO 390 M=1,NS
       IF(LL(M).NE.0) TT(M)=TT(M)+1.0
       IF(LL(M).NE.0.AND.LL(M).LE.15) A(L,M,I)=A(L,M,I)+1.0
  390  CONTINUE
       CALL FILEOP(4,F(1,I),1,1)
```

JUL 27 10:57 1976 B:FDA.F PAGE 2

```
      WRITE(6,100)(F(J,I),J=1,5)
  100 FORMAT(//7H FILE: ,5A4)
      WRITE(6,102)(J,J=1,15)
  102 FORMAT(/25H SITES:     TOT:    LENGTH:,15(I4,1H:))
      DO 290 J=1,NS
  290 WRITE(6,101)(FA(K,J),K=1,2),TT(J),(A(K,J,I),K=1,15)
  101 FORMAT(1X,2A4,I6,10X,15I5)
      DO 220 M=1,NS
      Q=0.0
      DO 220 J=1,15
      IF(TT(M).NE.0.0) Q=Q+A(J,M,I)/TT(M)
  220 A(J,M,I)=0.02*FLOAT(IFIX(50.0*Q+0.5))
  200 II=4
      WRITE(6,109)Z(IA),XX
  109 FORMAT(1H1//7H LEVEL:,F5.1,A1)
      WRITE(6,190)
  190 FORMAT(//38H CUMULATED FREQUENCY VS SERIES LENGTH:)
      WRITE(6,110)((FA(I,J),I=1,5),J=1,NS)
  110 FORMAT(/7H SITES:,5(5A4,3X))
      WRITE(6,111)
  111 FORMAT(1X)
      N=1
      DO 500 II=1,6
      C1=S1
      C2=S2
      DO 500 I=1,10
      DO 520 J=1,NS
      DO 530 L=2,15
  530 R(L,J)=C2
      DO 535 L=1,16,5
  535 R(L,J)=C1
      Q=0.02*FLOAT(51-N)
      IF(II.EQ.6) GO TO 520
      DO 540 L=1,15
      DO 540 K=1,NF
      IF(A(L,J,K).NE.Q) GO TO 540
      IF(R(L+1,J).EQ.X2) GO TO 540
      S=X1
      IF(MT(K).EQ.0) S=X2
      R(L+1,J)=S
  540 CONTINUE
  520 CONTINUE
      WRITE(6,150)(Q,(R(K,L),K=1,16),L=1,NS)
  150 FORMAT(1X,5(F7.2,16A1))
      IF(II.EQ.6) GO TO 580
      C1=S3
      C2=S4
  500 N=N+1
  580 L=0
      DO 590 I=1,4
      LL(I)=L
  590 L=L+5
      WRITE(6,160)((LL(L),L=1,4),I=1,NS)
  160 FORMAT(1X,5(3X,4I5))
      RETURN
      END
```

```
JUN 18 15:51 1976  B:FFA.F PAGE 1


C       "FFA" - FLOW FREQUENCY ANALYSIS.
C
        SUBROUTINE FFA(A,MK)
        DIMENSION A(50,5,10)
C
C       PARAMETERS:
C         A  - WORKING AREA,
C         MK - TRACE INDICATOR.
C
C       REQUIRED:
C         FILEOP  SETPAR  STAND1  ZEROS
C
        COMMON/PARAM1/IA(1),MT,NS,NT,NY,IC(3),FA(5,5)
        COMMON/WORK1/T(101)
        COMMON/WORK2/B(5,10)
        COMMON/WORK3/NF,F(5,10)
        COMMON/WORK4/MM(10)
        COMMON/WORK5/RYT(10)
        DATA CH1,CH2,CH3,CH4/4H-    ,4H+    ,4H     ,4HI    /
        DATA CH,CHS/4HH    ,4H*    /
        DATA X,E,LL/5.0,10.0,50/
        J=1
        DO 200 I=1,NF
        CALL STAND1(B(1,I),F(1,I),IA(1),J)
        MM(I)=MT
        CALL FILEOP(1,F(1,I),1,1)
        CALL SETPAR(1,1,IA(1),0,J)
        NST=NS*NT
        RYT(I)=FLOAT(NY*NT)
        CALL ZEROS(A(1,1,I),LL*NS)
        DO 220 K=1,NY
        CALL FILEOP(2,T(1),NST,1)
        DO 220 L=1,NS
        Q=B(L,I)
        DO 220 M=L,NST,NS
        N=1+IFIX(E*T(M)/Q)
        IF(N.LE.LL) A(N,L,I)=A(N,L,I)+1.0
  220 CONTINUE
        CALL FILEOP(4,F(1,I),1,1)
  200 J=4
        DO 300 N=1,NS
        WRITE(6,101)
  101 FORMAT(1H1//)
        WRITE(6,100)(FA(I,N),I=1,5),((F(I,J),I=1,2),J=1,NF)
  100 FORMAT(/6H SITE:,12X,5A4//9H FILE(S):,9X,10(2A4,1X))
        WRITE(6,110)(B(N,I),I=1,NF)
  110 FORMAT(/13H MEAN VALUES:,10F9.2)
        WRITE(6,115)(RYT(I),I=1,NF)
  115 FORMAT(13H TOT. COUNTS:,10I9)
        WRITE(6,120)
  120 FORMAT(/21H REDUCED-FLOW COUNTS:/)
        DO 415 K=1,LL
        Q=FLOAT(K)/E
  415 WRITE(6,105)Q,(A(K,N,J),J=1,NF)
  105 FORMAT(8X,F5.1,10I9)
        DO 410 J=1,NF
```

JUN 18 15:51 1976 B:FFA.F PAGE 2

```
      Q=0.0
      DO 410 I=1,LL
      Q=Q+A(I,N,J)/RYT(J)
  410 A(I,N,J)=Q
      WRITE(6,130)(FA(I,N),I=1,5),((F(I,J),I=1,2),J=1,NF)
      WRITE(6,131)(H(N,I),I=1,NF)
      WRITE(6,132)
  130 FORMAT(1H1//6H SITE:,12X,5A4//9H FILE(S):,9X,10(2A4,1X))
  131 FORMAT(13H MEAN VALUES:,10F9.2)
  132 FORMAT(/40H REDUCED FLOWS VS CUMULATED FREQUENCIES:/)
      L=LL
      DO 400 I=1,5
      S1=CH1
      S2=CH2
      DO 400 J=1,10
      DO 420 K=1,101
  420 T(K)=S1
      DO 430 K=1,101,10
  430 T(K)=S2
      DO 440 K=1,NF
      S=CHS
      IF(MM(K).EQ.0) S=CH
      M=IFIX(100.0*A(L,N,K)+0.5)+1
      IF(T(M).NE.CH) T(M)=S
  440 CONTINUE
      X=0.1*FLOAT(L)
      WRITE(6,150)X,(T(K),K=1,101)
  150 FORMAT(5X,F5.1,2X,101A1)
      S1=CH3
      S2=CH4
  400 L=L-1
      DO 450 K=1,101
  450 T(K)=CH1
      DO 460 K=1,101,10
  460 T(K)=CH2
      X=0.0
      WRITE(6,150)X,(T(K),K=1,101)
      DO 470 I=1,11
  470 T(I)=0.1*FLOAT(I-1)
      WRITE(6,160)(T(I),I=1,11)
  160 FORMAT(4X,11F10.1)
  300 CONTINUE
      RETURN
      END
```

```
C       "MAG" - MARKOVIAN GENERATION.
C
        SUBROUTINE MAG(A,NI,FN,MK)
        DIMENSION A(1),FN(1)
C
C       PARAMETERS:
C         A  - VECTOR OF GENERATION PARAMETERS:
C                MEAN(NS*NT)
C                SDEV(NS*NT)
C                B(NS*NS*NT)
C                A(NS*NS*NT)
C         NI - NUMBER OF INITIAL RANDOM NUMBER GENERATOR CYCLES,
C         FN - FILENAME OF GENERATED DATA,
C         MK - TRACE INDICATOR.
C
C       REQUIRED:
C         COPY    FILEOP  FGEN    PRINT   RANDNV
C         SETPAR  SETPTR  VECOP   ZEROS
C
        COMMON/WORK1/R(60)
        COMMON/WORK2/XP(5)
        COMMON/PARAM1/IA(8),FNA(5,5)
        COMMON/PARAM2/IB(8),FNB(5,5)
        COMMON/PARAM3/NS,NT,NY,ND,LA,LB,NSS,NST,NSSD,NSTD,NSST,NST1
        IF(IA(1).NE.3) CALL PRINT(4HGEN ,0,0.0,IA(1),-3)
        CALL SETPTR(IA(3),IA(4),IB(5),2)
        NST1=NST+1
        IF(NI.LE.0) GO TO 200
        DO 250 I=1,NI
  250   X=RANDNV(1.0,0.0)
  200   CALL ZEROS(XP(1),NS)
C
C         GENERATE SYNTHETIC RECORDS:
C
        IF(NY.EQ.0) GO TO 800
        CALL FILEOP(-1,FN(1),1,1)
        CALL COPY(IA(1),IB(1),33)
        IB(5)=NY
        CALL SETPAR(1,2,IB(1),0,1)
        DO 310 I=1,NY
        LB=NSTD+1
        DO 320 J=1,NST,NS
        LA=LB+NSST
        CALL FGEN(A(LA),A(LB),XP(1),R(J),NS,NS)
        CALL COPY(R(J),XP(1),NS)
  320   LB=LB+NSS
        CALL VECOP(A(1),A(NST1),R(1),NST,4)
        IF(MK.NE.0) CALL PRINT(0,-1,R(1),NS,NT)
  310   CALL FILEOP(3,R(1),NST,1)
        CALL FILEOP(4,FN(1),1,1)
  800   RETURN
        END
```

APR 30 14:59 1976 B:MAM.F PAGE 1

```
C       "MAM" - ESTIMATION OF GENERATION PARAMETERS.
C
        SUBROUTINE MAM(A,F,M,MK)
        DIMENSION A(1),F(1)
C
C       PARAMETERS:
C          A   - VECTOR OF PARAMETERS:
C                MEAN(NS*NT)
C                SDEV(NS*NT)
C                CORR-LAG-0(NS*NS*NT),   B(NS*NS*NT)
C                CORR-LAG-1(NS*NS*NT),   A(NS*NS*NT)
C          F   - INPUT-FILE-NAME,
C          M   - MODE,
C          MK  - TRACE INDICATOR.
C
C       REQUIRED:
C          CMAB     COPY     ECORR    PRINT    SETPTR   STAND
C
        COMMON/WORK1/XX(25)
        COMMON/WORK2/XN(25)
        COMMON/WORK3/YX(25)
        COMMON/PARAM1/IA(8),FA(25)
        COMMON/PARAM3/NS,NT,NY,ND,LA,LB,NSS,NST,LXX,NSTD,NSST,LYY
        CALL STAND(A(1),F(1),IA(1),1)
        CALL ECORR(A(1),A(2*NST+1),F(1),IA(1),1,M)
        CALL SETPTR(IA(3),IA(4),IA(5),2)
        LB=NSTD+1
        LA=LB+NSST
        IF(MK.NE.0) CALL PRINT(12H MATRIX: R0 ,12,A(LB),NSS,NT)
        IF(MK.NE.0) CALL PRINT(12H MATRIX: R1 ,12,A(LA),NSS,NT)
C
C          INITIALLY XN CONTAINS LAST-SEASON CORR-LAG-0 MATRIX:
C
  200   LXX=LB+(NT-1)*NSS
        CALL COPY(A(LXX),XN(1),NSS)
C
C         FOR EACH SEASON: XN IS TRANSFERED TO XX,
C           ACTUAL CORR-LAG-0 TO XN, ACTUAL CORR-LAG-1 TO YX,
C           ACTUAL CORR-LAG-0 IS REPLACED BY B-MATRIX,
C           ACTUAL CORR-LAG-1 IS REPLACED BY A-MATRIX:
C
        LXX=LA-1
        DO 300 LYY=LB,LXX,NSS
        LYX=LYY+NSST
        CALL COPY(XN(1),XX(1),NSS)
        CALL COPY(A(LYY),XN(1),NSS)
        CALL COPY(A(LYX),YX(1),NSS)
  300   CALL CMAB(XX(1),A(LYY),A(LYX),YX(1),NS,NS)
        IF(MK.EQ.0) GO TO 800
        CALL PRINT(12H MATRIX: A  ,12,A(LA),NSS,NT)
        CALL PRINT(12H MATRIX: B  ,12,A(LB),NSS,NT)
  800   IA(1)=3
        IA(2)=1
        IA(8)=1
        RETURN
        END
```

APR 30 15:01 1976 B:MYM.F PAGE 1

```
C       "MYM" - ESTIMATION OF AVERAGE GENERATION PARAMETERS.
C
        SUBROUTINE MYM(A,F,M,MK)
        DIMENSION A(1),F(1)
C
C       PARAMETERS:
C         A   - VECTOR OF PARAMETERS:
C               MEAN(NS*NT)
C               SDEV(NS*NT)
C               CORR-LAG-0(NS*NS*NT),   B(NS*NS*NT)
C               CORR-LAG-1(NS*NS*NT),   A(NS*NS*NT)
C         F   - INPUT-FILE-NAME,
C         M   - MODE,
C         MK  - TRACE INDICATOR.
C
C       REQUIRED:
C         CMAB     COPY     ECORR    PRINT    SETPTR   STAND
C         VECOP
C
        COMMON/PARAM1/IA(8),FA(25)
        COMMON/PARAM3/NS,NT,NY,LA,LB,LC,NSS,NST,LK,NSTD,NSST,LL
        COMMON/WORK1/AA(25)
        COMMON/WORK2/BB(25)
        DIMENSION FF(1)
        EQUIVALENCE(FF(1),FT)
        CALL STAND(A(1),F(1),IA(1),1)
        CALL ECORR(A(1),A(2*NST+1),F(1),IA(1),1,M)
        CALL SETPTR(IA(3),IA(4),IA(5),2)
        LB=NSTD+1
        LA=LB+NSST
        LC=NSTD+NSST
        IF(NT-1) 990,190,100
  100   FT=1.0/FLOAT(NT)
        LF=LB+NSS
        DO 110 LL=LF,LC,NSS
        LK=LL+NSST
        CALL VECOP(A(LB),A(LL),A(LB),NSS,1)
  110   CALL VECOP(A(LA),A(LK),A(LA),NSS,1)
        CALL VECOP(FF(1),A(LB),A(LB),NSS,5)
        CALL VECOP(FF(1),A(LA),A(LA),NSS,5)
  190   CALL COPY(A(LB),BB(1),NSS)
        IF(MK.NE.0) CALL PRINT(12H MATRIX: R0A,12,A(LB),NS,NS)
        IF(MK.NE.0) CALL PRINT(12H MATRIX: R1A,12,A(LA),NS,NS)
  200   CALL CMAB(A(LB),BB(1),AA(1),A(LA),NS,NS)
        IF(MK.EQ.0) GO TO 290
        CALL PRINT(12H MATRIX:"A"A,12,AA(1),NS,NS)
        CALL PRINT(12H MATRIX:"B"A,12,BB(1),NS,NS)
  290   DO 300 LL=LB,LC,NSS
        LK=LL+NSST
        CALL COPY(AA(1),A(LK),NSS)
  300   CALL COPY(BB(1),A(LL),NSS)
        IA(1)=3
        IA(2)=2
        IA(8)=1
  990   RETURN
        END
```

JUN 18 14:13 1976 S:OUT.F PAGE 1

```
C       "OUT" - PRINT GENERATED AND HISTORICAL STATISTICS.
C
        SUBROUTINE OUT(A,FN,MK)
        DIMENSION A(1),FN(1)
C
C       PARAMETERS:
C          A   - VECTOR OF STATISTICAL PARAMETERS:
C                MEAN(NS*NT)
C                SDEV(NS*NT)
C                SKEW(NS*NT)
C                KURT(NS*NT)
C                CORR-LAG-0(NS*NS*NT)
C                CORR-LAG-1(NS*NS*NT)
C          FN  - FILENAME OF STATISTICAL PARAMETERS.
C          MK  - INDICATOR.
C
C       REQUIRED:
C          COPY     FILEOP  SETFT  SETPAR
C
        DIMENSION T(2),Z(2,4),FF(13,2),X(7,4),Y(3)
        COMMON/PARAM1/I1,I2,NS,NT,NY,I6,I7,I8,FN1(5,5)
        COMMON/PARAM2/IB(8),FN2(25)
        COMMON/WORK2/F1(13),F2(13)
        COMMON/WORK1/R(60)
        COMMON/WORK3/D(5)
        COMMON/WORK4/V(5)
        EQUIVALENCE (FF(1,1),F1(1)),(FF(1,2),F2(1))
        DATA Z(1,1),Z(2,1)/4HM E ,4HA N /
        DATA Z(1,2),Z(2,2)/4HS D ,4HE V /
        DATA Z(1,3),Z(2,3)/4HS K ,4HE W /
        DATA Z(1,4),Z(2,4)/4HK U ,4HR T /
        DATA X(1,1),X(1,2),X(1,3),X(1,4)/4HH I ,4HM G ,4HY G ,4HO I /
        DATA X(2,1),X(2,2),X(2,3),X(2,4)/4HS T ,4HE N ,4HE N ,4HS A /
        DATA X(3,1),X(3,2),X(3,3),X(3,4)/4HO R ,4HE R ,4HE R ,4HG G /
        DATA X(4,1),X(4,2),X(4,3),X(4,4)/4HI C ,4HA T ,4HA T ,4HR E /
        DATA X(5,1),X(5,2),X(5,3),X(5,4)/4HA L ,4HE D ,4HE D ,4HG . /
        DATA X(6,1),X(6,2),X(6,3),X(6,4)/4H D A,4H D A,4H D A,4H D A/
        DATA X(7,1),X(7,2),X(7,3),X(7,4)/4H T A,4H T A,4H T A,4H T A/
        DATA Y(1),Y(2),Y(3)/4H. & ,4HA G ,4HG . /
        DATA T(1)/4HZERO/,T(2)/4HONE /,S/2H S/,P/2H P/
        IF(I1.NE.2) CALL PRINT(4HOUT ,0,0.0,I1,-2)
        CALL FILEOP(1,FN(1),1,1)
        CALL SETPAR(1,1,IB(1),2,5)
        NSS=NS*NS
        NST=NS*NT
        J=I2
        DO 220 I=1,2
        K=IABS(J)+1
        CALL COPY(X(1,K),FF(1,I),7)
        IF(J.LT.0) CALL COPY(Y(1),FF(3,I),3)
  220   J=IB(2)
        WRITE(6,100)((FF(J,I),J=1,7),I=1,2)
  100   FORMAT(1H1//28X,7A4,9X,7A4)
  230   WRITE(6,102)NY,I6,I7,(IB(J),J=5,7)
  102   FORMAT(19X,2(9X,3HNY=,I4,10X,1H(,I4,1H-,I4,1H)))
        WRITE(6,105)(I,(FN1(J,I),J=1,5),I=1,NS)
```

```
JUN 18 14:13 1976  B:OUT.F PAGE 2

  105 FORMAT((56X,1HS,I1,2H: ,5A4))
      CALL SETFT(F1(1),36H(**X,5X,*(7X,A2,I1),14X,5(7X,A2,I1)),36,1,NS)
      CALL SETFT
     1    (F2(1),40H(**X,A2,I2,1X,*F10.2,9X,A2,I2,1X,5F10.2),40,2,NS)
      N=1
C
C        PRINT MEAN, SDEV, SKEW, KURT:
C
      DO 3ØØ K=1,4
      CALL FILEOP(2,R(1),NST,1)
      WRITE(6,1Ø1)Z(1,K),Z(2,K)
  1Ø1 FORMAT(//56X,2A4,1H:/)
      WRITE(6,F1)((S,M,M=1,NS),L=1,2)
      J=1
      DO 2ØØ I=1,NT
      NN=N+NS-1
      JJ=J+NS-1
      WRITE(6,F2)P,I,(A(M),M=N,NN),P,I,(R(M),M=J,JJ)
      J=J+NS
  2ØØ N=N+NS
  3ØØ CONTINUE
C
C        PRINT CORR-LAG=Ø,1:
C
      DO 4ØØ I=1,2
      CALL SETFT
     1    (F2(1),41H(**X,A2,I1,2X,*F1Ø.4,**X,A2,I1,2X,5F1Ø.4),41,3,NS)
      WRITE(6,11Ø)T(I)
  11Ø FORMAT(//46X,4HLAG ,A4,20H CORRELATION MATRIX:/)
      DO 4ØØ J=1,NT
      WRITE(6,111)J
  111 FORMAT(58X,2HF=,I2)
      WRITE(6,F1)((S,K,K=1,NS),L=1,2)
      CALL FILEUP(2,R(1),NSS,1)
      DO 41Ø K=1,NS
      KK=K
      IF (I.NE.1) KK=NS
      CALL SETFT(F2(1),Ø,Ø,4,KK)
      CALL SETF1(F2(1),Ø,Ø,5,NS-KK+1)
      M=K
      NN=M+K-1
      DO 43Ø L=1,KK
      Q(L)=A(NN)
  44Ø V(L)=R(M)
      M=M+NS
  43Ø NN=NN+NS
  41Ø WRITE(6,F2)S,K,(Q(L),L=1,KK),S,K,(V(L),L=1,KK)
  4ØØ N=N+NSS
      CALL FILEOP(4,FN(1),2,1)
      RETURN
      END
```

JUL 27 12:37 1976 B:RWC.F PAGE 1

```
C       "RWC" - READ/WRITE STATISTICS & PARAMETERS.
C
        SUBROUTINE RWC(MD,A,FO,MK)
        DIMENSION A(1),FO(1)
C
C       PARAMETERS:
C         MD - REQUIRED OPERATION:
C              MD=1 READING FROM FILE
C              MD=2 WRITING INTO FILE
C         A  - VECTOR OF STATISTICAL PARAMETERS:
C              MEAN(NS*NT)
C              SDEV(NS*NT)
C              SKEW(NS*NT)
C              KURT(NS*NT)
C              CORR-LAG-0(NS*NS*NT)
C              CORR-LAG-1(NS*NS*NT)
C         FO - FILENAME,
C         MK - TRACE INDICATOR.
C
        COMMON/PARAM1/I1,I2,I3,I4,I5,I6,I7,I8,T(25)
        COMMON/PARAM3/NS,NT,NY,ND,N1,N2,NSS,NST,N3,NSTD,NSST,N
        DIMENSION IA(8)
        EQUIVALENCE (IA(1),I1)
        I=MD
        IF(I.NE.1) I=-1
        CALL FILEOP(I,FO(1),1,1)
        CALL SETPAR(1,MD,IA(1),11,1)
        IF(I1.LT.2.OR.I1.GT.4) CALL PRINT(4HRWC ,0,0.0,I1,-234)
        CALL SETPTR(I3,I4/I8,I5,I8)
        GO TO (200,202,204,204),I1
  200   STOP
  202   N1=4
        N2=2*NT
        N3=NSS
        GO TO 210
  204   N1=2
        N2=2
        N3=NSST*ND
  210   L=NSTD
        K=1
        N=1
        DO 211 I=1,2
        DO 212 J=1,N1
        CALL FILEOP(MD+1,A(K),L,1)
        K=K+L
  212   L=L*N
        N=ND
        N1=N2
  211   L=N3
        CALL FILEOP(4,FO(1),1,1)
        RETURN
        END
```

```
C      "SEL" - SELECTION OF DATA RECORDS FROM BASIC FILES.
C
       SUBROUTINE SEL(FN,MK)
       DIMENSION FN(1)
C
C      PARAMETERS:
C        FN - OUTPUT FILENAME,
C        MK - TRACE INDICATOR.
C
C      REMARK:
C        ORDERING OF DATA AFTER SELECTION FROM BASIC FILES:
C          YEAR.1-PERIOD.1-SITE.1
C                         -SITE.2
C                         -......
C                 -PERIOD.2-SITE.1
C                         -SITE.2
C          ......-........-......
C
C      REQUIRED:
C        COPYX   FILEOP  PRINT   SETPAR
C
       COMMON/PARAM2/IDF,IDS,NS,NT,NY,IF,IT,M,T(5,5)
       COMMON/WORK1/Q(60)
       COMMON/WORK2/R(38)
       DIMENSION IB(8),LK(4)
       EQUIVALENCE (IB(1),IDF)
       DATA LK(1),LK(2),LK(3),LK(4)/1,4,12,36/
       NT=LK(M)
       NY=IT-IF+1
       NST=NS*NT
C
C        OPEN FILES:
C
       DO 200 I=1,NS
       IF(MK.NE.0) CALL PRINT(T(1,I),20,0,0,0)
   200 CALL FILEOP(1,T(1,I),5,10+I)
       CALL FILEOP(-1,FN(1),1,1)
       IDS=0
       CALL SETPAR(1,2,IB(1),0,1)
C
C        READ DATA RECORDS, REORDER, AND WRITE INTO OUTPUT FILE:
C
       DO 201 I=IF,IT
       FY=I
       DO 202 J=1,NS
   203 CALL FILEOP(2,R(1),2,10+J)
       K1=IFIX(R(2))
       CALL FILEOP(2,R(3),LK(K1),10+J)
       IF(R(1)-FY)203,700,900
   700 IF(K1-M)203,750,720
   750 CALL COPYX(R(3),Q(J),1,NS,1,NT)
       GO TO 202
   720 KN=LK(K1)/L
       IF(KN*L.NE.LK(K1)) GO TO 203
       DO 230 K1=J,NST,NS
       S=0.0
```

MAY 4 11:32 1976 B:SEL.F PAGE 2

```
      DO 240 K2=1,KN
      K=K+1
  240 S=S+R(K)
  230 Q(K1)=S/FLOAT(KN)
  202 CONTINUE
      IF(MK.NE.0) CALL PRINT(0,-1,Q(1),NS,NT)
  201 CALL FILEOP(3,Q(1),NST,1)
C
C        CLOSE FILES:
C
      CALL FILEOP(4,FN(1),1,1)
      DO 204 I=1,NS
  204 CALL FILEOP(4,T(1,I),5,10+I)
      RETURN
C
C        EXCEPTIONS:
C
  900 WRITE(6,110)T(1,J),T(2,J),R(1),R(2)
  110 FORMAT(//30H INCORRECT DATA IN BASIC FILE:,2A4,3X,I4,I3)
      STOP 00000
      END
```

APR 26 16:18 1976 B:TRF.F PAGE 1

```
C       "TRF" - TRANSFORMATIONS OF DATA.
C
        SUBROUTINE TRF(MT,FI,FO,MK)
        DIMENSION FI(1),FO(1)
C
C       PARAMETERS:
C         MT - TYPE OF TRANSFORMATION:
C              MT=0 NO TRANSFORMATION
C              MT=1 LOG TRANSFORMATION
C              MT=2 SQRT TRANSFORMATION
C              MT=3 CBRT TRANSFORMATION
C              MT=-1 EXP TRANSFORMATION
C              MT=-2 SQR TRANSFORMATION
C              MT=-3 CUB TRANSFORMATION
C         FI - FILENAME OF ORIGINAL DATA,
C         FO - FILENAME OF TRANSFORMED DATA,
C         MK - TRACE INDICATOR.
C
C       REQUIRED:
C         FILEOP  PRINT   SETPAR
C
        COMMON/PARAM2/IDF,IDS,NS,NT,NY,IF,IT,MM,FN(25)
        COMMON/WORK1/R(60)
        DIMENSION IB(8)
        EQUIVALENCE (IB(1),IDF)
    300 CALL FILEOP(1,FI(1),1,1)
        CALL SETPAR(1,1,IB(1),0,1)
        MI=-MT
        NST=NS*NT
        CALL FILEOP(-1,FO(1),1,2)
        CALL SETPAR(2,2,IB(1),IDF,1)
        DO 200 I=1,NY
        CALL FILEOP(2,R(1),NST,1)
        DO 400 K=1,NST
        X=R(K)
        IF(MT) 500,700,600
    500 GO TO(510,520,530),MI
    510 X=EXP(X)
        GO TO 400
    520 X=X*X
        GO TO 400
    530 X=X*X*X
        GO TO 400
    600 GO TO(610,620,630),MT
    610 X=ALOG(X)
        GO TO 400
    620 X=SQRT(X)
        GO TO 400
    630 X=EXP(ALOG(X)/3.0)
    400 R(K)=X
    700 IF(MK.NE.0) CALL PRINT(4H TRF,4,R(1),NS,NT)
    200 CALL FILEOP(3,R(1),NST,2)
        CALL FILEOP(4,FI(1),1,1)
        CALL FILEOP(4,FO(1),1,2)
        RETURN
        END
```

APR 14 16:10 1976 Y:CHECK.F PAGE 1

```
C      "CHECK" - COMPARISON OF DESCRIPTION RECORDS.
C
       SUBROUTINE CHECK(IA,K,L)
       DIMENSION IA(1)
C
C      PARAMETERS:
C        IA- DESCRIPTION RECORD AREA,
C        L - STARTING INDEX,
C        K - FINAL INDEX.
C
       COMMON/PARAM1/IB(8)
       IF(K.GT.L) GO TO 900
       DO 200 I=K,L
       IF(IA(I).NE.IB(I)) GO TO 900
  200  CONTINUE
       RETURN
  900  WRITE(6,100)
  100  FORMAT(///22H  DESCRIPTION RECORDS:)
       WRITE(6,110)(I,IA(I),IB(I),I=1,8)
  110  FORMAT(1X,I3,2H: ,2I8)
       STOP
       END
```

MAY 4 09:27 1976 D:CMAB.F PAGE 1

```
C      "CMAB" - COMPUTATION OF "A" AND "B" MATRICES.
C
       SUBROUTINE CMAB(XX,YY,XY,YX,N,M)
       DIMENSION XX(1),YY(1),XY(1),YX(1)
C
C      PARAMETERS:
C        XX - MATRIX N*N,
C        YY - MATRIX M*M,
C        XY - MATRIX N*M,
C        YX - TRANSP(XY), MATRIX M*N,
C        N  - SIZE OF XX MATRIX,
C        M  - SIZE OF YY MATRIX.
C
C      REMARKS:
C        MATRIX YY IS REPLACED BY COMPUTED MATRIX B.
C        MATRIX XY IS REPLACED BY COMPUTED MATRIX A.
C        MATRIX XX IS REPLACED BY COMPUTED MATRIX C.
C        INSTEAD OF XY, TRANSP(YX) IS USED.
C
C        MINV - IBM/360 SSP SUBROUTINE.
C
C      REQUIRED:
C        MINV     MMULT    PRINT    TRNG     VECOP
C
       COMMON/WORK4/IW(5)
       COMMON/WORK5/JW(5)
C
C        INVERT XX-MATRIX:
C
       CALL MINV(XX(1),N,D,IW(1),JW(1))
C
C        CHECK DETERMINANT:
C
       IF(ABS(D).GE.1E-10) GO TO 200
       CALL PRINT(36H "CMAB" - INCORRECT MATRIX INVERSION,36,0,-1,0)
       STOP
C
C        COMPUTE A-MATRIX, A=YX*INV(XX) :
C
   200 CALL MMULT(YX(1),XX(1),XY(1),M,N,N)
C
C        COMPUTE C-MATRIX, C=YY-A*XY :
C
       CALL MMULT(XY(1),YX(1),XX(1),M,N,-M)
       CALL VECOP(YY(1),XX(1),XX(1),M*M,2)
C
C        COMPUTE B-MATRIX FROM B*TRANSP(B)=C:
C
       CALL TRNG(XX(1),YY(1),M)
       RETURN
       END
```

```
APR 29 15:55 1976  Y:COPY.F PAGE 1

C      "COPY" - COPYING VECTORS.
C
       SUBROUTINE COPY(A,B,N)
       DIMENSION A(1),B(1)
C
C      PARAMETERS:
C        A - SOURCE VECTOR,
C        B - RESULTING VECTOR,
C        N - SIZE OF "A" AND "B" VECTORS.
C
       IF(N) 100,200,300
  100  M=-N
       DO 150 I=1,M
  150  B(I)=-A(I)
       RETURN
  300  DO 350 I=1,N
  350  B(I)=A(I)
  200  RETURN
       END
```

```
APR 14 14:38 1976  W:COPYX.F PAGE 1

C       "COPYX" - COPYING SUBVECTORS.
C
        SUBROUTINE COPYX(A,B,K,L,M,N)
        DIMENSION A(1),B(1)
C
C       PARAMETERS:
C          A - SOURCE VECTOR,
C          B - RESULTING VECTOR,
C          K - SUBVECTOR LENGTH IN A,
C          L - SUBVECTOR LENGTH IN B,
C          M - LENGTH OF COPIED SUBVECTOR,
C          N - NUMBER OF SUBVECTORS IN A AND B.
C
        I=1
        JJ=L*N
        DO 200 J=1,JJ,L
        CALL COPY(A(I),B(J),M)
  200   I=I+K
        RETURN
        END
```

APR 29 10:26 1976 U:CORR.F PAGE 1

```
C       "CORR" - CALCULATION OF CORRELATION COEFFICIENTS.
C
        SUBROUTINE CORR(X,Y,R,N1,N2,N3,M)
        DIMENSION X(1),Y(1),R(1)
C
C       PARAMETERS:
C         X   - VECTOR OF X DATA(N1*N3),
C         Y   - VECTOR OF Y DATA(N2*N3),
C         R   - VECTOR OF CORR-COEFF(N1*N2*N3),
C         N1 - SIZE OF SUBVECTOR IN X,
C         N2 - SIZE OF SUBVECTOR IN Y,
C         N3 - NUMBER OF SUBVECTORS IN X AND Y,
C         M   - MODE.
C
C       REMARK:
C         CORR-COEFF ARE STORED IN THE FOLLOWING ORDER:
C         X(1)*Y(1), X(1)*Y(2), X(1)*Y(3), ...
C
        IF(M) 200,300,400
  200   STOP
  300   IK=-N1
        JK=-N2
        KK=0
        DO 350 K=1,N3
        IK=IK+N1
        JK=JK+N2
        DO 350 I=1,N1
        II=IK+I
        XX=X(II)
        DO 350 J=1,N2
        JJ=JK+J
        KK=KK+1
  350   R(KK)=R(KK)+XX*Y(JJ)
        RETURN
  400   KK=N1*N2*N3
        XX=X(1)
        DO 450 I=1,KK
        YY=XX*R(I)
        IF(YY.GT.1.0) YY=1.0
  450   R(I)=YY
        RETURN
        END
```

```
APR 30 10:00 1976  D:ECORR.F PAGE 1

C       "ECORR" - ESTIMATION OF CORRELATION COEFFICIENTS.
C
        SUBROUTINE ECORR(A,B,F,N,K,M)
        DIMENSION A(1),B(1),F(1),N(1)
C
C       PARAMETERS:
C         A - VECTOR OF STANDARDIZATION PARAMETERS:
C             MEAN(NS*NT)
C             SDEV(NS*NT)
C         B - VECTOR OF CORRELATION COEFFICIENTS:
C             CORR-LAG-0(NS*NS*NT)
C             CORR-LAG-1(NS*NS*NT)
C         F - INPUT-FILE-NAME,
C         N - DESCRIPTION RECORD AREA,
C         K - CYCLIC SHIFT,
C         M - MODE.
C
C        REQUIRED:
C          COPY     CORR     FILEOP   SETPAR   SETPTR   VECOP
C          ZEROS
C
      COMMON/PARAMS/NS,NT,NY,NK,NSK,M1,M2,NST,M3,M4,NSST,L
      COMMON/WORK1/R(70)
      DIMENSION FF(1)
      EQUIVALENCE (FF(1),FY)
      CALL FILEOP(1,F(1),1,1)
      CALL SETPAR(1,1,N(1),0,1)
      CALL SETPTR(N(3),N(4),N(5),K)
      M1=NSK+1
      M2=NST+1
      M3=NSST+1
      CALL ZEROS(B(1),2*NSST)
      IF(M) 120,120,130
  130 J=M1
      I=M2
  140 J=J-NS
      IF(J) 120,120,150
  150 I=MAX0(1,I-NS)
  170 L=I+NST
      CALL VECOP(A(I),A(L),R(J),NS,3)
      GO TO 140
  120 DO 190 I=1,NY
      CALL FILEOP(2,R(M1),NST,1)
      CALL VECOP(A(1),A(M2),R(M1),NST,3)
      CALL CORR(R(M1),R(M1),B(1),NS,NS,NT,0)
      IF(M.EQ.0.AND.I.EQ.1) GO TO 190
      CALL CORR(R(1),R(M1),B(M3),NS,NS,NT,0)
  190 CALL COPY(R(M2),R(1),NSK)
      CALL FILEOP(4,F(1),1,1)
      FY=1.0/FLOAT(NY)
      CALL CORR(FF(1),B(1),B(1),NS,NS,NT,1)
      IF(M.EQ.0) FY=1.0/FLOAT(NY-1)
      CALL CORR(FF(1),B(M3),B(M3),NS,NS,NT,1)
      RETURN
      END
```

```
C       "FGEN" - COMPUTATION OF Y = A * X + B * RANDOM.
C
        SUBROUTINE FGEN(A,B,X,Y,N,M)
        DIMENSION A(1),B(1),X(N),Y(M)
C
C       PARAMETERS:
C         A - MATRIX (M*N),
C         B - MATRIX (M*M),
C         X - VECTOR (N),
C         Y - VECTOR (M),
C         N - SIZE OF "X", "A",
C         M - SIZE OF "Y", "B", "A".
C
C       REQUIRED:
C         RANDNV
C
        COMMON/WORK3/Q(20)
C
C         GENERATE RANDOM NORMAL (M,1) VECTOR:
C
        DO 200 I=1,M
  200   Q(I)=RANDNV(1.0,0.0)
C
C         COMPUTE VECTOR Y:
C
        DO 300 I=1,M
        S=0.0
        JA=I
        DO 400 J=1,N
        S=S+X(J)*A(JA)
  400   JA=JA+M
        JB=I
        DO 500 J=1,M
        S=S+Q(J)*B(JB)
  500   JB=JB+M
  300   Y(I)=S
        RETURN
        END
```

APR 14 16:09 1976 Y:FILEOP.F PAGE 1

```
C      "FILEOP/PDP" - FILE OPERATION AND EXCEPTIONS.
C
       SUBROUTINE FILEOP(M,A,L,N)
       DIMENSION A(L)
C
C      PARAMETERS:
C        M - MODE OF OPERATION:
C             M=0 SET EXCEPTIONS,
C             M=1 OPEN FILE,
C             M=2 READ RECORD,
C             M=3 WRITE RECORD,
C             M=4 CLOSE FILE;
C        A - FILENAME(M=1) OR RECORD AREA(M=2,3)
C        L - LENGTH OF A
C        N - PROGRAM (LOGICAL) FILE NUMBER/FILE-OPERATION TRACE MODE:
C             M=0 AND N=0 NO TRACE
C                     N=-1 OPEN-CLOSE FILE TRACE
C                     N=1 OPEN-CLOSE FILE, READ-WRITE RECORD TRACE
C
C      REMARK:
C        FILE OPERATIONS AND FILE EXCEPTIONS FOLLOW PDP-UNIX FORTRAN
C        POSSIBLITIES.
C
C        IERROR - PDP FORTRAN FUNCTION,
C        SETFIL - PDP FORTRAN SUBROUTINE.
C
       IF(M) 201,202,200
C
C        SET EXCEPTIONS:
C
   202 MK=N
C
C        MK - FILE-OPERATION-TRACE INDICATOR.
C
       IF(IERROR(102).NE.0) GO TO 902
       IF(IERROR(103).NE.0) GO TO 903
       IF(IERROR(104).NE.0) GO TO 904
       RETURN
C
C        FILE OPERATIONS:
C
   201 M=-M
   200 GO TO(210,220,230,240),M
C
C        OPEN FILE:
C
   210 CALL SETFIL(N,A)
       IF(MK.EQ.0) RETURN
       WRITE(6,151)N,A(1),A(2)
   151 FORMAT("OPEN FILE: ",I2,3X,2A4)
       RETURN
C
C        READ RECORD:
C
   220 READ(N)(A(I),I=1,L)
       IF(MK.LE.0) RETURN
```

```
APR 14 16:09 1976  Y:FILEOP.F PAGE 2

      WRITE(6,152)N,L
  152 FORMAT(" READ FILE:",I2,I5)
      RETURN
C
C        WRITE RECORD:
C
  230 WRITE(N)(A(I),I=1,L)
      IF(MK.LE.0) RETURN
      WRITE(6,153)N,L
  153 FORMAT("WRITE FILE:",I2,I5)
      RETURN
C
C        CLOSE FILE:
C
  240 ENDFILE(N)
      IF(MK.EQ.0) RETURN
      WRITE(6,154)N
  154 FORMAT("CLOSE FILE:",I2)
      RETURN
C
C        EXCEPTIONS:
C
  902 WRITE(6,102)
  102 FORMAT(/" CANNOT CREATE OUTPUT FILE")
      GO TO 990
  903 WRITE(6,103)
  103 FORMAT(/" CANNOT OPEN INPUT FILE")
      GO TO 990
  904 WRITE(6,104)
  104 FORMAT(/" EOF ON INPUT FILE")
  990 WRITE(6,199)N
  199 FORMAT(I5)
      STOP
      END
```

```
APR 14 14:43 1976  W:MATOP.F PAGE 1

C       "MATOP" - MATRIX OPERATIONS.
C
        SUBROUTINE MATOP(A,B,C,M,N,K)
        DIMENSION A(1),B(1),C(1)
C
C       PARAMETERS:
C          A - FIRST ARGUMENT MATRIX,
C          B - SECOND ARGUMENT MATRIX,
C          C - RESULTING MATRIX,
C          M - NUMBER OF ROWS,
C          N - NUMBER OF COLUMNS,
C          K - OPERATION.
C
        MN=M*N
        L=0
        GO TO(100,200,300,400),K
C
C          ADDITION: C=A+TRANSP(B)
C
   100  DO 111 J=1,N
        DO 111 I=J,MN,N
        L=L+1
   111  C(L)=A(L)+B(I)
        RETURN
C
C          SUBTRACTION: C=A-TRANSP(B)
C
   200  DO 222 J=1,N
        DO 222 I=J,MN,N
        L=L+1
   222  C(L)=A(L)-B(I)
        RETURN
C
C          ADDITION: C=B+A(1)*IDENT
C
   300  DO 303 I=1,MN
   303  C(I)=B(I)
   330  AA=A(1)
        J=M+1
        DO 333 I=1,MN,J
   333  C(I)=C(I)+AA
        RETURN
C
C          SUBTRACTION: C=B-A(1)*IDENT
C
   400  DO 404 I=1,MN
   404  C(I)=-B(I)
        GO TO 330
        END
```

APR 30 09:48 1976 Y:MMULT.F PAGE 1

```
C       "MMULT" - MATRIX MULTIPLICATION: C=A*B,C=A*TRANSP(B),C=TRANSP(A)*B
C
        SUBROUTINE MMULT(A,B,C,N,M,L)
        DIMENSION A(1),B(1),C(1)
C
C       PARAMETERS:
C         A - ARGUMENT MATRIX,
C         B - ARGUMENT MATRIX,
C         C - RESULTING MATRIX,
C         N - SIZE OF A MATRIX (N,M),
C         M - SIZE OF A,B MATRICES,
C         L - SIZE OF B MATRIX (M,L).
C
C       REMARK:
C         IF N<0 INSTEAD OF A-MATRIX TRANSP(A) IS USED,
C         IF L<0 INSTEAD OF B-MATRIX TRANSP(B) IS USED.
C
        LL=IABS(L)
        IF(L) 910,900,920
   900  STOP
   910  MK=1
        MB=LL
        GO TO 930
   920  MK=M
        MB=1
   930  NN=IABS(N)
        IF(N) 940,900,950
   940  NK=M
        NA=1
        GO TO 990
   950  NK=1
        NA=N
   990  IC=0
        IK=1
        DO 100 LB=1,LL
        JK=1
        DO 200 J=1,NN
        IC=IC+1
        JA=JK
        IB=IK
        R=0.0
        DO 300 I=1,M
        R=R+A(JA)*B(IB)
        JA=JA+NA
   300  IB=IB+MB
        JK=JK+NK
   200  C(IC)=R
   100  IK=IK+MK
        RETURN
        END
```

```
APR 30 11:03 1976  U:MNSD.F PAGE 1

C       "MNSD" - ESTIMATION OF MEAN,SDEV,SKEW,KURT COEFFICIENTS.
C
        SUBROUTINE MNSD(X,A,L,M)
        DIMENSION X(1),A(1)
C
C       PARAMETERS:
C         X - VECTOR OF DATA,
C         A - VECTOR OF COEFFICIENTS,
C         L - SIZE OF "X", "A" VECTORS,
C         M - MODE:
C              M<0 - SKEW & KURT,
C              M=0 - MEAN & SDEV,
C              M>0 - FINAL  CALC.
C
        IF(M) 200,300,400
  200   DO 250 I=1,L
        XX=X(I)
        YY=XX*XX
        A(I)=A(I)+XX*YY
        J=I+L
  250   A(J)=A(J)+YY*YY
        RETURN
  300   DO 350 I=1,L
        Y=X(I)
        A(I)=A(I)+Y
        J=I+L
  350   A(J)=A(J)+Y*Y
        RETURN
  400   Y=X(1)
        DO 450 I=1,L
        Z=Y*A(I)
        A(I)=Z
        J=I+L
  450   A(J)=SQRT(Y*A(J)-Z*Z)
        RETURN
        END
```

APR 14 14:26 1976 U:PRINT.F PAGE 1

```
C      "PRINT" - GENERAL PRINT SUBROUTINE.
C
       SUBROUTINE PRINT(T,L,A,M,N)
       DIMENSION T(1),A(1)
C
C      PARAMETERS:
C        T - VECTOR CONTAINING HEADER TEXT,
C        L - NUMBER OF CHARACTERS IN T-VECTOR,
C        A - VECTOR TO BE PRINTED IN MATRIX FORM,
C        M - NUMBER OF ROWS,
C        N - NUMBER OF COLUMNS.
C
C      REMARK:
C        IT IS ASSUMED THAT ONE STORAGE UNIT CONTAINS 4 CHARACTERS.
C
       IF(L) 200,200,900
  900  K=(L+3)/4
       WRITE(6,100)(T(I),I=1,K)
  100  FORMAT(/25A4)
  200  IF(N) 310,300,320
  320  WRITE(6,110)(I,I=1,N)
  110  FORMAT(/(6X,14(I7,1H:)))
       IF(M.LT.0) GO TO 400
       MN=M*N
       DO 250 I=1,M
  250  WRITE(6,120)I,(A(K),K=I,MN,M)
  120  FORMAT(1X,I3,2H: ,14F8.2/(6X,14F8.2))
       RETURN
  310  N=-N
       WRITE(6,130)T(1),M,N
  130  FORMAT(///12H SUBROUTINE ,A4,10HVIOLATION:,I1,1H/,I1)
  302  STOP
  300  IF(M) 302,399,301
  301  WRITE(6,150)T(1),M
  150  FORMAT(/A4,I2)
  399  RETURN
  400  M=-M
       DO 450 I=1,M
       I2=I*N
       I1=I2-N+1
  450  WRITE(6,120)I,(A(K),K=I1,I2)
       RETURN
       END
```

```
APR 14 16:09 1976  Y:RANDNV.F PAGE 1

C       "RANDNV/PDP" - NORM.DISTR.RANDOM NUMBER GENERATOR.
C
        FUNCTION RANDNV(SD,AM)
C
C       PARAMETERS:
C         SD - STANDARD DEVIATION,
C         AM - MEAN VALUE.
C
C       REMARK:
C         RANDV - PDP FORTRAN FUNCTION.
C
        A=0.0
        DO 50 I=1,12
   50   A=A+RANDV(0)
        RANDNV=(A-6.0)*SD+AM
        RETURN
        END
```

APR 14 14:27 1976 U:READ.F PAGE 1

```
C       "READ" - READING PARAMETERS FOR "MAIN".
C
        SUBROUTINE READ(IT,A,LI,W)
        DIMENSION A(1)
C
C       PARAMETERS:
C         IT - PARAMETER TYPE,
C         A  - RESULTING VECTOR OF PARAMETERS,
C         LI - NUMBER OF ELEMENTS,
C         W  - REQUIRED LEADING STRING.
C
        EQUIVALENCE (K,T)
        L=IT*LI
        DO 200 I=1,L,IT
        IF(IT.EQ.5) GO TO 300
        IF(IT.EQ.3) GO TO 400
        READ(5,100)S,K
  100   FORMAT(A4,I4)
        A(I)=T
  250   IF(S.EQ.W) GO TO 200
        WRITE(6,190)S,W
  190   FORMAT(///12H PARAMETERS?,A4,1H(,A4,1H))
        STOP
  300   K=I+4
        READ(5,110)S,(A(J),J=I,K)
  110   FORMAT(6A4)
        GO TO 250
  400   READ(5,120)S,A(1),A(2),K
  120   FORMAT(A4,A3,A1,I1)
        A(3)=T
        GO TO 250
  200   CONTINUE
        RETURN
        END
```

MAY 4 09:58 1976 W:SETFT.F PAGE 1

```
C      "SETFT/POP" - CREATION OF DYNAMIC FORMATS FOR "OUT".
C
       SUBROUTINE SETFT(F,T,L,M,N)
       DIMENSION F(1),T(1)
C
C      PARAMETERS:
C        F - VECTOR FOR CREATED FORMAT,
C        T - BASIC FORM OF CREATED FORMAT,
C        L - NUMBER OF CHARACTERS IN "T",
C        M - IDENTIFIER OF CREATED FORMAT (M > 0),
C        N - VARIANT OF CREATED FORMAT (N=1,...,5).
C
C      CREATED FORMATS:
C        M=1: (**X,5X,*(7X,A2,I1),14X,5(7X,A2,I1))
C        M=2: (**X,A2,I2,1X,*F10.2,9X,A2,I2,1X,5F10.2)
C        M=3: (**X,A2,I1,2X,*F10.4,**X,A2,I1,2X,5F10.4)
C
C      REMARK:
C        IT IS ASSUMED THAT ONE STORAGE UNIT CONTAINES 4 CHARACTERS.
C        IF THIS ASSUMPTION IS NOT TRUE SEGMENT SHOULD BE CHANGED.
C
C      REQUIRED:
C        COPY
C
       DIMENSION A(5),B(5),C(5),D(5)
       DATA A(1),A(2),A(3),A(4),A(5)/4H(41X,4H(31X,4H(21X,4H(11X,4H(01X/
       DATA B(1),B(2),B(3),B(4),B(5)/4H1(7X,4H2(7X,4H3(7X,4H4(7X,4H5(7X/
       DATA C(1),C(2),C(3),C(4),C(5)/4HX,1F,4HX,2F,4HX,3F,4HX,4F,4HX,5F/
       DATA D(1),D(2),D(3),D(4),D(5)/4H,09X,4H,19X,4H,29X,4H,39X,4H,49X/
       IF(L) 201,202,203
  201  STOP
  203  J=(L+3)/4
       CALL COPY(T(1),F(1),J)
  202  IF(M) 251,252,253
  251  STOP
  253  GO TO(301,302,303,304,305),M
  301  F(3)=B(N)
       GO TO 303
  302  F(4)=C(N)
  303  F(1)=A(N)
  252  RETURN
  304  F(4)=C(N)
       RETURN
  305  F(6)=D(N)
       RETURN
       END
```

```
APR 14 14:45 1976  W:SETPAR.F PAGE 1

C      "SETPAR" - READING & WRITING THE DESCRIPTION RECORDS.
C
       SUBROUTINE SETPAR(N,M,IA,L,K)
       DIMENSION IA(1)
C
C      PARAMETERS:
C        N - FILE-NUMBER,
C        M - OPERATION (M=1 READ, M=2 WRITE),
C        IA- DESCRIPTION RECORD AREA,
C        L - FILE-TYPE,
C        K - CHECK-CODE.
C
C      REQUIRED:
C        CHECK    FILEOP
C
       IF(M.EQ.2) IA(1)=L
       CALL FILEOP(M+1,IA(1),8,N)
       IF(IA(1).NE.L) CALL CHECK(IA(1),1,0)
       CALL FILEOP(M+1,IA(9),IA(3)*5,N)
       GO TO (310,320,330,340,350,360),K
  320  I=0
       GO TO 300
  330  I=4
       GO TO 300
  340  I=5
  300  DO 305 J=3,7
       IF(J.NE.I) CALL CHECK(IA(1),J,J)
  305  CONTINUE
  310  RETURN
  350  I=4
  355  CALL CHECK(IA(1),3,I)
       RETURN
  360  CALL CHECK(IA(1),3,3)
       CALL CHECK(IA(1),6,7)
       RETURN
       STOP
       END
```

```
APR 14 16:09 1976  Y:SETPTR.F PAGE 1

C       "SETPTR" - COMPUTATION OF THE "COMMON/PARAM3/" PARAMETERS.
C
        SUBROUTINE SETPTR(N1,N2,N3,N4)
C
        COMMON/PARAM3/NS,NT,NY,ND,NSD,NTD,NSS,NST,NSSD,NSTD,NSST,NSSDD
        NS=N1
        NT=N2
        NY=N3
        ND=N4
        NSD=NS*ND
        NTD=NT*ND
        NSS=NS*NS
        NST=NS*NT
        NSSD=NSS*ND
        NSTD=NST*ND
        NSST=NSS*NT
        NSSDD=NSD*NSD
        RETURN
        END
```

APR 30 09:51 1976 D:STAND.F PAGE 1

```
C       "STAND" - ESTIMATION OF STANDARDIZATION PARAMETERS.
C
        SUBROUTINE STAND(A,F,N,K)
        DIMENSION A(1),F(1),N(1)
C
C       PARAMETERS:
C         A - VECTOR OF RESULTING PARAMETERS,
C         F - INPUT-FILENAME,
C         N - DESCRIPTION RECORD AREA,
C         K - CYCLIC SHIFT.
C
C       REQUIRED:
C         COPY     FILEOP   MNSD     SETPAR   SETPTR    ZEROS
C
        COMMON/PARAM3/NS,NT,NY,NK,NSK,NTK,NSS,NST,NSSK,NSTK,M1,M2
        COMMON/WORK1/R(70)
        DIMENSION FF(1)
        EQUIVALENCE (FF(1),FY)
        CALL FILEOP(1,F(1),1,1)
        CALL SETPAR(1,1,N(1),0,1)
        CALL SETPTR(N(3),N(4),N(5),K)
        FY=1.0/FLOAT(NY)
        M1=NSK+1
        M2=NST+1
        CALL ZEROS(A(1),2*NST)
        DO 100 I=1,NY
        CALL FILEOP(2,R(M1),NST,1)
        CALL MNSD(R(M1),A(1),NST,0)
    100 CALL COPY(R(M2),R(1),NSK)
        CALL FILEOP(4,F(1),1,1)
        CALL MNSD(FF(1),A(1),NST,1)
        RETURN
        END
```

MAY 21 09:37 1976 D:STAND1.F PAGE 1

```
C       "STAND1" - ESTIMATION OF GLOBAL MEAN VALUES.
C
        SUBROUTINE STAND1(A,F,N,M)
        DIMENSION A(1),F(1),N(1)
C
C       PARAMETERS:
C         A - VECTOR OF MEAN VALUES,
C         F - FILE-NAMF,
C         N - DESCRIPTION-RECORD AREA,
C         M - CHECK-CODE.
C
C       REQUIRED:
C         FILEOP  SETPAR  VECOP   ZERUS
C
        COMMON/WORK1/NS,NY,NST,R(60)
        CALL FILEOP(1,F(1),1,1)
        CALL SETPAR(1,1,N(1),0,M)
        NS=N(3)
        NY=N(5)
        NST=NS*N(4)
        CALL ZEROS(A(1),NS)
        DO 100 I=1,NY
        CALL FILEOP(2,R(1),NST,1)
        DO 100 J=1,NST,NS
  100   CALL VECOP(R(J),A(1),A(1),NS,1)
        R(1)=1.0/FLOAT(NY*N(4))
        CALL VECOP(R(1),A(1),A(1),NS,5)
        CALL FILEOP(4,F(1),1,1)
        RETURN
        END
```

```
APR 14 14:25 1976  WITRNG.F PAGE 1

C      "TRNG" - SOLUTION OF THE EQUATION B * TRANSP(B) = C.
C
       SUBROUTINE TRNG(C,B,N)
       DIMENSION B(1),C(1)
C
C      PARAMETERS:
C        C - SYMMETRIC MATRIX N*N OF DATA,
C        B - MATRIX N*N OF RESULTS,
C        N - SIZE OF "C", "B" MATRICES.
C
C      REMARKS:
C        "C" AND "B" MATRICES CANNOT OVERLAP.
C        EIGEN - IBM/360 SSP SUBROUTINE.
C
       NN=N*N
C
C        CHANGE MATRIX "C" FROM SQUARE TO TRIANGULAR:
C
       L=1
       DO 200 I=1,N
       IL=L
       DO 210 JL=L,NN,N
       C(JL)=0.5*(C(IL)+C(JL))
  210  IL=IL+1
  200  L=L+N+1
       K=0
       L=0
       DO 250 I=1,NN,N
       J=L+I
       DO 260 JL=I,J
       K=K+1
  260  C(K)=C(JL)
  250  L=L+1
C
C        COMPUTE EIGENVALUES AND EIGENVECTORS:
C
       CALL EIGEN(C(1),B(1),N,0)
C
C        CHECK EIGENVALUES AND COMPUTE "B" MATRIX:
C
       L=0
       K=0
       DO 300 I=1,N
       L=L+I
       Q=C(L)
       IF(Q.LT.-0.001) WRITE(6,100)I,Q
  100  FORMAT(/23H SUBROUTINE TRNG-EIGEN:,I6,1H:,E10.3)
       IF(Q.LT.0.0) Q=0.0
       Q=SQRT(Q)
       DO 300 J=1,N
       K=K+1
  300  B(K)=B(K)*Q
       RETURN
       END
```

```
APR 14 14:40 1976  W:VECOP.F PAGE 1

C      "VECOP" - VECTOR OPERATIONS.
C
       SUBROUTINE VECOP(A,B,C,N,M)
       DIMENSION A(1),B(1),C(1)
C
C      PARAMETERS:
C        A - FIRST ARGUMENT VECTOR,
C        B - SECOND ARGUMENT VECTOR,
C        C - RESULTING VECTOR,
C        N - SIZE OF A,B,C VECTORS,
C        M - OPERATION.
C
       GO TO (100,200,300,400,500),M
C
C        VECTOR ADDITION:
C
  100  DO 111 I=1,N
  111  C(I)=A(I)+B(I)
       RETURN
C
C        VECTOR SUBTRACTION:
C
  200  DO 222 I=1,N
  222  C(I)=A(I)-B(I)
       RETURN
C
C        STANDARDIZATION:
C
  300  DO 333 I=1,N
  333  C(I)=(C(I)-A(I))/B(I)
       RETURN
C
C        DESTANDARDIZATION:
C
  400  DO 444 I=1,N
  444  C(I)=B(I)*C(I)+A(I)
       RETURN
C
C        SCALING:
C
  500  AA=A(1)
       DO 555 I=1,N
  555  C(I)=B(I)*AA
       RETURN
       END
```

```
APR 29 14:49 1976  Y:ZEROS.F PAGE 1

C       "ZEROS" - MOVING ZEROS TO A VECTOR.
C
        SUBROUTINE ZEROS(A,N)
        DIMENSION A(N)
C
C       PARAMETERS:
C         A - VECTOR-NAME,
C         N - LENGTH OF "A".
C
        DO 100 I=1,N
    100 A(I)=0.0
        RETURN
        END
```

References

Aitchison, J. and J.C. Brown (1957), *The lognormal distribution*, Cambridge.

Anderson, T.W. (1958), *Introduction to multivariate statistical analysis*, New York-London.

Bernier, J. (1971), Modèles probabilistes à variables hydrologiqes multiples et hydrologie synthétique, *Proc. Int. Symposium on Mathematical Models in Hydrology*, IAHS Publication, No. 100.

Fiering, M.B. and B.B. Jackson (1971), Synthetic streamflow, *Water Resources Monograph*, No. 1, American Geophysical Union. Washington, D.C.

Finyi, G., E. Todini, and J.R. Wallis (1974), Comment upon multivariate synthetic hydrology, *Proc. of Workshop on Mathematical Models in Hydrology*, Pisa.

Kaczmarek, Z. (1970), *Metody statystyczne w hydrologü i meteorologü* (Statistical methods in hydrology and meteorology), (in polish), WKiL, Warsaw.

Mandelbrot, B.B. and J.R. Wallis (1969), Computer experiments with Fractional Gaussian Noises, *Water Resour. Research*, 5(1).

Matalas, N.C. (1967), Mathematical assessment of synthetic hydrology, *Water Resour. Research*, 3(4).

Mejia, J.M., I. Rodriguez-Iturbe, and D.R. Daudy (1972), Streamflow Simulation. The broken line process as a potential model for hydrologic simulation, *Water Resour. Research*, 8(4).

Mejia, J.M., J. Rouselle (1976), Disaggregation models in hydrology revisited, *Water Resour. Research*, 12(2).

O'Connell, P.E. (1972), General Report, *Proc. of Int. Symposium on Uncertainties in Hydrologic and Water Resource Systems* Tucson, Arizona.

O'Connell, P.E. (1973), *Multivariate synthetic hydrology: a correction*, J. Hydraul, Dir. Amer. Soc. Civil Eng., HY 12.

Svanidze, G.G. (1964), *Principles of estimating river flow regulation by the Monte Carlo method*, Tbilisi, (in Russian).

Thomas, H.A. and M.B. Fiering (1962), Mathematical synthesis of streamflow sequences for the analysis of river basins by simulation, Ch. 12 of Maas, A. et al, *Design of water resources systems*, Harvard Univ. Press.

Trykozko, E. (1973), *Typical distributions of hydrometeorological phenomena*, (in polish), Ph.D. Thesis, Warsaw Polytechnic University.

Valencia, D.R. and J.C. Schaake Jr. (1972), A disaggregation model for time series analysis and synthesis, *MIT Report*, No. 149, June 1972.

Yevjevich, V. (1964), Fluctuations of wet and dry years, Part II, Analysis by serial correlation, *Colorado State University Hydrology Papers*, No. 4.

Yevjevich, V. (1966), Stochastic problems in design of reservoirs, *Water Research*, John Hopkins Press, Baltimore, Maryland.

Young, G.K. and W.C. Pisano (1968), *Operational hydrology using residuals*, J. Hydraul, Dir. Amer. Soc. Civil Eng., 94(HY4).